

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la aprobación del título de:  
Ingeniera e Ingeniero de Sistemas**

**TEMA:  
DESARROLLO DE UN BOTÓN DE SOCORRO EN UNA PLATAFORMA  
MÓVIL**

**AUTORA Y AUTOR:  
GABRIELA ESTEFANIA GUAMÁN GUAMÁN  
FREDDY STEVEN ORQUERA LUNA**

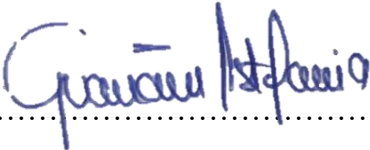
**TUTOR:  
JULIO RICARDO PROAÑO ORELLANA**

**Quito, agosto del 2020**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Gabriela Estefanía Guamán Guamán con documento de identificación N° 1719257071 y Freddy Steven Orquera Luna con documento de identificación N° 1724484108, manifestamos la voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: **“DESARROLLO DE UN BOTÓN DE SOCORRO EN UNA PLATAFORMA MÓVIL”**, mismo que ha sido desarrollado para optar por el título de: INGENIERA E INGENIERO DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



Gabriela Estefanía Guamán Guamán

Cédula: 1719257071

Fecha: Quito, agosto 2020



Freddy Steven Orquera Luna

Cédula: 1724484108

Fecha: Quito, agosto 2020

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, **DESARROLLO DE UN BOTÓN DE SOCORRO EN UNA PLATAFORMA MÓVIL** realizado por Gabriela Estefanía Guamán Guamán y Freddy Steven Orquera Luna, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto 2020



---

Julio Ricardo Proaño Orellana

C.I: 0103909412

## **DEDICATORIA**

El presente proyecto de titulación lo dedico primeramente a Dios quien me dio la fuerza principal para superar todas las dificultades y continuar mis metas propuestas, a mis dos ángeles que me dejaron un dolor muy grande, pero recuerdos que siempre estarán en mi corazón.

A mi padre Gustavo Guamán y a mi madre Lida Guamán que creyeron siempre en mí y con esfuerzo, trabajo y mucho sacrificio hicieron todo lo posible para la culminación de mis estudios, debo decir que todo lo que hoy soy es gracias a ellos.

A mi esposo Galo Jaramillo quien con su apoyo y mucho amor supo estar a mi lado inclusive en los momentos más difíciles, a mis hijos Scarlett y Mateo mis mejores regalos de Dios, son y serán siempre el principal motor de mi vida, los que me ayudan a ser cada vez mejor. Les dedico este logro que es un ejemplo de superación. Los amo

Finalmente agradezco de manera muy especial a Julissa, Mishell, Mayerly y Freddy que formaron parte de todo el proceso brindándome un apoyo incondicional, sin importar el momento.

Gabriela Estefanía Guamán Guamán

## **DEDICATORIA**

El presente proyecto se lo dedico de manera muy especial a mi madre Silvia Orquera quien siempre me apoyo y alentó para seguir adelante convirtiéndose en un cimiento principal en mi vida personal y profesional, a mi padre Gustavo Orquera por haberme dado todo lo necesario para culminar mis estudios.

A mi hermano Andrés Guamán a quien quiero mucho y agradezco por siempre brindarme su apoyo incondicional, llegando a formar parte de este logro profesional. Demostrándole que el trabajo duro siempre da su fruto.

Finalmente, doy gracias a toda mi familia y amigos que siempre estuvieron presentes y nunca permitieron que me rinda.

Freddy Steven Orquera Luna

## **AGRADECIMIENTO**

Primeramente, agradecemos a la Universidad Politécnica Salesiana por brindarnos la oportunidad de formarnos como profesionales en la carrera de Ingeniería de Sistemas, así como también a cada docente que nos brindó su tiempo y conocimiento durante todo el proceso.

A nuestro tutor de tesis Julio Proaño quien estuvo pendiente en todo el proceso brindándonos su asesoría continua en el desarrollo de la aplicación como en la revisión del documento, invirtiendo tiempo y dedicación.

Gabriela Estefanía Guamán Guamán

Freddy Steven Orquera Luna

## INDICE

INTRODUCCIÓN .....	1
Justificación.....	3
Objetivos .....	4
Objetivo General.....	4
Objetivos Específicos.....	4
Capítulo 1 .....	6
1. Marco Teórico .....	6
1.1. Inseguridad .....	6
1.1.1. Situación social .....	6
1.1.2. Económico y cultural.....	6
1.2. Delincuencia.....	7
1.2.1. Delincuencia organizada.....	7
1.2.2. Delincuencia juvenil.....	8
1.2.3. Delincuencia tecnológica o cibernética .....	8
1.3. Sistemas de atención de emergencias .....	9
1.3.1. Sistema ECU 911 .....	9
1.3.2. Amiga, Ya No Estás Sola (AYNES) .....	10
1.4. Aplicaciones Móviles.....	12
1.4.1. Sistema operativo móvil .....	12
1.4.2. Arquitectura de Conexión Cliente - Servidor .....	15
1.5. Aplicaciones híbridas.....	17
1.5.1. Lenguaje Híbrido.....	18
1.5.2. Frameworks.....	20
1.5.3. Ventajas y desventajas.....	22
1.6. SCRUM.....	23
1.6.1. Características .....	23
1.6.2. Ventajas y Desventajas .....	24
1.6.3. Componentes.....	24
1.6.4. Funcionamiento .....	26
Capítulo 2 .....	35
2.1. Análisis de Factibilidad.....	35
2.1.1. Factibilidad Técnica .....	35
2.1.2. Factibilidad Económica .....	38
2.1.3. Factibilidad Operacional .....	40
2.2. Análisis de Requerimientos.....	41
2.3. Alcance del Prototipo.....	43
Capítulo 3 .....	44
3.1. Implementación de la metodología.....	44
3.2. Descripción general de requerimientos.....	48
3.2.1. Características del Prototipo .....	48
3.2.2. Perfiles de usuarios .....	49
3.2.3. Restricciones .....	49
3.2.4. Superposiciones y dependencias .....	50
3.3. Requisitos de rendimiento.....	51
3.4. Interfaces .....	52
3.4.2. Usuario .....	52
3.4.3. Comunicación .....	54
3.5. Requerimientos.....	55
3.5.2. Requerimientos funcionales.....	55

3.5.3.	Requerimientos no funcionales .....	59
3.6.	Diseño .....	61
3.6.1.	Diagrama de caso de uso .....	61
3.6.2.	Diagrama de secuencia .....	63
3.6.3.	Diagrama de base de datos .....	78
3.6.4.	Interfaces .....	80
3.6.5.	Diagrama de Clases .....	93
3.6.6.	Diagrama de Componentes .....	95
Capítulo 4	.....	97
4.1.	Implementación .....	97
4.1.1.	Creación de proyecto Firebase .....	97
4.1.2.	Conexión con la base de datos .....	99
4.2.	Implantación .....	101
Capítulo 5	.....	105
5.1.	Métricas de Evaluación .....	105
5.1.1.	Lista de métricas .....	105
5.1.2.	Esquema de puntuación .....	105
5.2.	Pruebas .....	106
5.2.1.	Pruebas de funcionamiento .....	106
5.2.1.	Pruebas de sistema .....	114
5.3.	Resultados .....	116
CONCLUSIONES	.....	121
RECOMENDACIONES	.....	122
LISTAS DE REFERENCIAS	.....	123



## INDICE DE FIGURAS

Figura 1 Proceso de funcionamiento de un llamado de emergencia .....	11
Figura 2 Capas que componen la plataforma Android .....	13
Figura 3 Capas del sistema operativo IOS .....	14
Figura 4 Modelo arquitectura Cliente-Servidor .....	16
Figura 5 Modelo P2P en el que todos los terminales operan como Cliente/Servidor .....	17
Figura 6 Esquema de trabajo donde se detalla el funcionamiento de una aplicación Híbrida.....	20
Figura 7 Capas Flutter con sus respectivos contenidos.....	21
Figura 8 Secuencia de la funcionalidad en la metodología Scrum. ....	26
Figura 9 Puntos involucrados para la realización de historias de usuario. ....	29
Figura 10 Estructura de una tabla Kanban .....	31
Figura 11 Proceso generalmente usado en reuniones de revisión de Sprint. ....	33
Figura 12 Estado visible de actividades para todos los usuarios .....	47
Figura 13 Representación de la guía para llegar a la interfaz final. ....	53
Figura 14 Representación de la conexión de un servidor a todos los usuarios. ....	55
Figura 15 Diagrama de caso de uso de la aplicación móvil .....	62
Figura 16 Diagrama de secuencia de registro de usuario.....	65
Figura 17 Diagrama de secuencia de autenticación de usuario.....	66
Figura 18 Diagrama de secuencia visualizar o registrar contacto. ....	68
Figura 19 Diagrama de secuencia visualizar o registrar un mensaje.....	70
Figura 20 Diagrama de secuencia emergencia de nivel rojo.....	72
Figura 21 Diagrama de secuencia emergencia nivel amarillo.....	73
Figura 22 Diagrama de secuencia emergencia nivel verde.....	75
Figura 23 Diagrama de secuencia perfil personal.....	77
Figura 24 Diagrama de secuencia de emergencias recibidas. ....	78
Figura 25 Diagrama base de datos del sistema de botón de pánico.....	79
Figura 26 Pantalla principal de la aplicación. ....	81
Figura 27 Pantalla de registro de usuario nuevo.....	82
Figura 28 Pantalla de perfil de usuario. ....	83
Figura 29 Pantalla de editar datos de usuario.....	84
Figura 30 Pantalla de visualización de contactos registrados. ....	85
Figura 31 Pantalla de registro de contacto. ....	86
Figura 32 Pantalla que muestra los niveles de emergencia. ....	87
Figura 33 Interfaz nivel rojo.....	88
Figura 34 Interfaz nivel amarillo. ....	89
Figura 35 Interfaz de nivel verde.....	90
Figura 36 Interfaz de mensajería .....	91
Figura 37 Interfaz de nuevo mensaje .....	92
Figura 38 Interfaz de mapa y alertas de emergencia. ....	93
Figura 39 Diagrama de clase aplicación “SAFE” .....	95
Figura 40 Diagrama de componentes aplicación “SAFE” .....	96
Figura 41 Designación de nombre del proyecto.....	97
Figura 42 Herramientas de Google Analytics. ....	98
Figura 43 Proyecto creado.....	98
Figura 44 Registro de app en Firebase.....	100

Figura 45 Archivo de configuraciones copiado en app. ....	100
Figura 46 Archivo build.gradle modificado y sincronizado. ....	101
Figura 47 Generar apk de la aplicación ....	102
Figura 48 Ubicación del apk en el dispositivo móvil ....	103
Figura 49 Aviso de aplicación desconocida ....	103
Figura 50 Aviso de aplicación desconocida ....	104
Figura 51 Prueba de registro en aplicación SAFE.....	107
Figura 52 Prueba de registro de información en la base de datos ....	108
Figura 53 Prueba de login en aplicación SAFE.....	109
Figura 54 Prueba de contactos en aplicación SAFE.....	110
Figura 55 Prueba de revisión de información en la base de datos.....	110
Figura 56 Prueba de perfil de usuario en aplicación SAFE ....	111
Figura 57 Prueba de semaforización en aplicación SAFE ....	112
Figura 58 Prueba de Mapa visualizar en aplicación SAFE.....	113
Figura 59 Prueba de Mapa con envío de alerta en aplicación SAFE.....	114
Figura 60 Grafica de fallas .....	118
Figura 61 Grafica de uso de CPU .....	119
Figura 62 Grafica de rendimiento de la aplicación.....	119

## INDICE DE TABLAS

Tabla 1 Características KOTLIN .....	18
Tabla 2 Ventajas y desventajas de aplicaciones híbridas .....	22
Tabla 3 Ventajas y Desventajas de Scrum .....	24
Tabla 4 Ejemplo de Pila de Producto.....	28
Tabla 5 Requerimientos de Hardware.....	36
Tabla 6 Requerimientos de software.....	36
Tabla 7 Recurso Humano .....	37
Tabla 8 Insumos y Servicios .....	38
Tabla 9 Costos Hardware .....	39
Tabla 10 Costos Software .....	39
Tabla 11 Total en Recursos .....	39
Tabla 12. Datos del Personal .....	42
Tabla 13 Roles en el proyecto .....	44
Tabla 14 Pila de Product Backlog .....	45
Tabla 15 Planificación de Product Backlog .....	47
Tabla 16 Perfil Colaborador. ....	49
Tabla 17 Perfil Suscriptor .....	49
Tabla 18 Registro de usuario.....	56
Tabla 19 Consultar datos de usuario.....	56
Tabla 20 Registro de contactos.....	57
Tabla 21 Consulta de Contactos de Emergencia .....	57
Tabla 22 Mensajería personalizada o predefinida .....	58
Tabla 23 Geo localización.....	59
Tabla 24 Flexibilidad .....	59
Tabla 25 Seguridad .....	60
Tabla 26 Mantenibilidad .....	60
Tabla 27 Usabilidad .....	61
Tabla 28 Ficha de caso de uso de la aplicación móvil.....	62
Tabla 29 Descripción de campos.....	79
Tabla 30 Descripción de clases de aplicación móvil .....	93
Tabla 31 Valores de medición.....	106
Tabla 32 Características dispositivo 1 .....	115
Tabla 33 Características dispositivo 2 .....	115
Tabla 34 Descripción del usuario .....	116
Tabla 35 Resultados de interacción en dispositivos físicos .....	117
Tabla 36 Crash en versiones.....	118

## **RESUMEN**

La inseguridad en general se ha convertido en uno de los principales problemas de la ciudad de Quito. Existen varios tipos de emergencias como maltrato familiar, maltrato infantil, robos en locales comerciales, secuestros, acosos, etc. Por otro lado, la poca eficacia a la atención de los llamados de emergencia por parte de la policía ha generado en los ciudadanos un efecto de inseguridad constante dentro y fuera de su domicilio. En la actualidad los dispositivos móviles han brindado un impacto positivo en nuestra sociedad, su uso hoy en día no tiene límite de edad, lo que permite una fácil adaptación a aplicaciones con finalidades ilimitadas. El desarrollo del presente trabajo se centra en la seguridad de los ciudadanos en la ciudad de Quito, y tiene como objetivo principal mejorar la eficacia en la atención a los llamados de emergencia mediante el desarrollo de un botón de socorro.

La aplicación es multiplataforma y brinda al usuario un libre acceso desde cualquier teléfono inteligente. El sistema permite notificar una emergencia a través de un mensaje de alerta a los contactos registrados. La información personal es almacenada en la base de datos en tiempo real la cual permite recopilar y mostrar la información en cualquier momento logrando una agilidad y automatización en el proceso de interacción con el usuario.

## **ABSTRACT**

Insecurity in general has become one of the main problems of the city of Quito. There are several types of emergencies such as family abuse, child abuse, robbery in local stores, kidnappings, harassment, etc. On the other hand, the ineffectiveness of responding to emergency calls by the police has generated a constant insecurity effect on citizens both inside and outside their homes. Currently mobile devices have had a positive impact on our society, their use nowadays has no age limit, allowing easy adaptation to applications with unlimited purposes. The development of this work focuses on the safety of citizens in the city of Quito, and has as its main objective to improve the effectiveness of responding to emergency calls by developing a help button.

The application is cross-platform and gives the user access from any smartphone. The system allows to make a emergency notification through an alert message to registered contacts. Personal information is stored in a database prepared for real-time transactions, which allows the information to be collected and displayed at any time, achieving agility and automation in the user interaction process.

## INTRODUCCIÓN

Al analizar los indicadores de violencia y delincuencia que expone el Instituto Nacional de Estadística y Censos se puede observar un incremento en la delincuencia y violencia en el 2018. Esto genera un efecto de inseguridad de la población en general. La sociedad debe reaccionar frente a estas conductas erróneas, sin embargo, el concepto de protección debe ampliarse.

En la ciudad de Quito, la unidad policial ha tomado como medida de seguridad la implementación de un botón de pánico comprometiéndose con una reacción rápida a sucesos que se produzcan en el barrio correspondiente. Sin embargo, han existido inconvenientes como el costo, capacitaciones inconclusas o no realizadas, inexistencia de comunicación telefónica en el domicilio o negocio, entre otras. Todos estos inconvenientes provocan una atención poco eficiente, ya que el usuario no tiene una manera fácil y rápida de conectarse o comunicarse a una entidad de auxilio. (Bravo, 2019)

Por otro lado, el uso de la tecnología móvil se ha introducido paulatinamente en la sociedad, sus avances técnicos y mejoras tanto en software como hardware lo han convertido en una evolución atractiva generando una inclusión en la vida cotidiana de las personas, pues muchos de los usuarios lo utilizan como recurso para sus negocios. La evolución del mercado es constante, debido a esto incrementa la competencia y obligan a las empresas a realizar continuamente innovaciones. (Zamora Delgado, 2019)

Este trabajo se enfoca en brindar apoyo al usuario, a través de un prototipo de aplicación móvil como botón de socorro, el cual consiste en el envío de un mensaje (predefinido o personalizado) de alerta de emergencia junto con la ubicación a sus contactos previamente registrados. La aplicación será capaz de ejecutarse en cualquier sistema operativo, brindando de tal forma comodidad y confianza al usuario. Por otra

parte, en el ámbito de desarrollo, la aplicación brinda un ahorro de costo y tiempo para posteriores innovaciones, de igual forma se obtendrá un mercado más amplio.

## **Justificación**

La aplicación móvil tiene como finalidad responder a una necesidad de auxilio con una respuesta rápida, es fundamental fomentar la relación de la seguridad personal con la tecnología, sobre todo cuando se trata de tomar medidas en caso de emergencias.

La ciudadanía al contar con un aplicativo destinado a su seguridad y a una comunicación rápida en caso de emergencia con su contacto seleccionado cumpliendo de esta manera la finalidad del objetivo. Para los clientes el llevar el dispositivo en todo momento no afecta su rutina y cuenta con un apoyo de seguridad sin necesidad de predecir el tipo de emergencia que le puede suceder durante el día, brindando de este modo tranquilidad al usuario.

El uso del aplicativo permite solicitar auxilio de manera virtual cuando es imposible la movilidad o se encuentre sujeto a un encierro que le impida acudir físicamente a pedir ayuda. La ventaja de esta aplicación es que el usuario solo debe disponer de un dispositivo móvil y se encontrara disponible para todos los usuarios sin costo o inconvenientes en la adaptación del manejo de la aplicación.



## **Objetivos**

Los ciudadanos serán beneficiados al igual que las entidades de apoyo o control de delitos como la policía, esto permitirá informar en tiempo real sobre el llamado de emergencia y la ubicación del usuario tanto a la parte policial como a los contactos predefinidos.

### **Objetivo General**

Desarrollar una aplicación móvil que facilite la comunicación de una emergencia en tiempo real a contactos previamente registrados.

### **Objetivos Específicos**

Diseñar un prototipo en una plataforma móvil que permita una interacción entre cliente y servidor con la finalidad de establecer un aviso instantáneo de la situación en la que se encuentra el usuario.

Analizar y diseñar una interfaz intuitiva y de fácil uso para obtener un alto grado de usabilidad.

Realizar pruebas de funcionamiento a la aplicación prototipo con la finalidad de comprobar y validar el cumplimiento de las especificaciones dadas.

El documento está dividido en cinco capítulos, que se detallan a continuación.

El primer capítulo contiene el marco teórico donde se incluirán conceptos y definiciones que abordan la investigación e intervinieron en el desarrollo del mismo.

Además, se especifica la metodología utilizada en el desarrollo del prototipo.

El segundo capítulo contiene el análisis de factibilidad, donde se detalla los puntos económicos, operacionales y técnicos en este último apartado se lo analiza tanto en software como en hardware.

El tercer capítulo contiene la metodología y el diseño, donde se detallan los requerimientos del usuario, interfaces de prototipo y diagramas necesarios para plasmar el proceso del desarrollo del sistema, así como funcionamiento.

El cuarto capítulo contiene la implementación e implantación del prototipo, donde se describe las técnicas utilizadas y pasos a seguir para la ejecución del proyecto. Este incluye la creación y conexión con la base de datos.

El quinto capítulo detalla la evaluación y resultados obtenidos a lo largo del desarrollo del prototipo, donde mediante métricas específicas se evalúa y se realiza pruebas en los diferentes módulos.

Finalmente, conclusiones y recomendaciones obtenidas tras la culminación del proyecto.

## **Capítulo 1**

### **1. Marco Teórico**

En el presente capítulo se detalla la recopilación de referencias teóricas que permiten plantear el problema de la investigación para continuamente ofrecer una solución.

#### **1.1. Inseguridad**

Según García Ocanto “La inseguridad es un flagelo que atañe al ciudadano en general y lo convierte en un ser cercado, limitado en su libertad, auto vigilado, temeroso y en constante sensación de acoso”.(García Ocanto et al., 2003,p.2)

A continuación, se detallan las causas que promueven la inseguridad.

##### **1.1.1. Situación social**

La situación social se basa en conductas relevantes que provocan inseguridad a la sociedad en general.(Torres López, Munguía Cortés, & Torres Valdovinos, 2018) en su estudio argumentan que el desempleo se considera como un tiempo desperdiciado generando una serie de necesidades que conllevan a una emoción negativa provocando depresión y en ocasiones hasta el suicidio, afectando de este modo el ámbito familiar y social.

##### **1.1.2. Económico y cultural**

El artículo de (Gil Villa, 2019) plantea las posibles razones por las cuales en América Latina el ser humano elige delinquir y afrontar el castigo. El estudio se basa en los historiales de los individuos que ingresan a prisión, pues una de las circunstancias que provocan esta conducta es tener que subsistir y mantener el hogar con una precaria situación económica por lo esto conduce a un comportamiento violento y delictivo. (Quintana Zurita, Rosero Moncayo, Serrano Salgado, & Pimentel Bolaños, 2014) en su estudio realizado analiza las acciones que provocan el uso de poder de un hombre hacia una mujer. Quintana deduce que

los celos, el consumo de alcohol, el engaño y los problemas económicos son los antecedentes con mayor porcentaje de generar la violencia familiar. Sin embargo, en su análisis basado en encuestas realizadas a mujeres, se obtuvo como resultado que el estado socioeconómico se ubica en tercer lugar con un porcentaje del (36.4%), debido a la discordia que presenta con el conviviente al momento de evidenciar necesidades en el hogar, generando una discusión que produce actos violentos. Según INEC en el Ecuador en los 12 últimos meses del año 2019 el tipo de violencia en el ambiente social ha variado entre psicológica, física y sexual donde el machismo es una de las justificaciones más usadas para cubrir su delito, generando una cifra alta de violencia y la mayoría no es denunciada (INEC, 2019). Según el análisis de Lema las cifras de mujeres que no formalizan la denuncia es del 52.5% dando como explicación el sufrimiento de sus hijos, estado económico o integridad familiar.(Lema Peñafiel, 2019)

## **1.2. Delincuencia**

Según (Figueroa, 2009) la delincuencia es una inconformidad ante la sociedad y es conocida comúnmente como un acto delictivo o conjunto de infracciones que dañan el orden público y requiere de atención para la búsqueda de una corrección, siempre y cuando se respete los derechos de las personas que se encuentren bajo este estado jurídico.

Este fenómeno da un énfasis a la conducta antisocial del individuo, en el cual se observan irregularidades en el carácter, personalidad y relación interpersonal. Existen diversos tipos de delincuencia que se especifican a continuación.

### **1.2.1. Delincuencia organizada**

Es la evolución de la delincuencia común, este tipo de infracción posee una estrategia o perfeccionamiento en las operaciones delictivas como soborno,

violencia o acciones ilegales, que se ejecutan de manera que afectan a la humanidad. Una particularidad especial es la distribución de actividades y jerarquías existentes donde encabeza un jefe y obedecen los intermediarios.(Murrieta López, 2013)

La delincuencia organizada posee varias características que interpretan de una mejor manera su definición:

- Comportamiento y disciplina ilegal.
- Ganancias rápidas sin trabajo o inversión alguna.
- Venta de bienes, servicios, transacciones ilegales.

### **1.2.2. Delincuencia juvenil**

Muñoz hace referencia a un sector específico que son los jóvenes, por consiguiente, se deduce que se relaciona con las personas que no han alcanzado la mayoría de edad al cometer el delito. Este término no tiene un significado concreto, existen diferentes puntos de vista, como el determinar una edad específica para identificarlo como delincuente juvenil o el hecho de establecer cuál es la conducta que lo considera a un joven como delincuente.

Ante la sociedad estos problemas se relacionan con la falta de oportunidades desde la niñez, a pesar de ello también influyen características como el estado socio-económico, desintegración familiar y maltrato infantil siendo este último uno de los principales puntos de referencia para la generación de la delincuencia. (Muñoz Caiza, 2019)

### **1.2.3. Delincuencia tecnológica o cibernética**

La delincuencia tecnológica se la define como una conducta de ámbito virtual, misma que utiliza la tecnología para establecer una comunicación con el ciberespacio y conjuntamente cometer actos delictivos que en muchas ocasiones

alcanzan a dañar la integridad personal, intimidad o libertad de expresión en la ciudadanía.(Pascual, 2013)

En este sentido las unidades de investigación son dotadas con medios tecnológicos que a través de investigaciones especializadas y el equipo necesario sean capaces de realizar un trabajo sigiloso y efectivo, que aporte de manera significativa un proceso de enjuiciamiento de manera rápida, logrando así una lucha contra la ciberdelincuencia. Es evidente que la sociedad posee un temor al enfrentarse a este tipo de conducta, debido a la confidencialidad del o los delincuentes, por lo cual genera una incertidumbre y desconfianza aun mayor al combatir contra un individuo no real. (Rayón Ballesteros & Gómez Hernández, 2014)

Ante los eventos mencionados se implementó ciertas medidas que brinden apoyo a la ciudadanía, estas se definen a continuación.

### **1.3. Sistemas de atención de emergencias**

Los sistemas de emergencia ya sea para atención personal o colectiva tienen como finalidad brindar una respuesta rápida y eficiente que permita de este modo gestionar de forma adecuada todas las solicitudes de auxilio que se puedan presentar sin llegar a tener una demanda de peticiones sin atención. (Castelo Rivas, Castelo Rivas, & Rodríguez Díaz, 2016)

A continuación, se mencionan algunos sistemas de atención de emergencias implementadas en Ecuador.

#### **1.3.1. Sistema ECU 911**

El gobierno ecuatoriano implemento un sistema nacional de emergencia ciudadana que garantiza seguridad y atención permanente ante incidentes en salud, incendios, robos entre otros. Además, se implementó un número fácil de recordar donde se atiende las llamadas de emergencia continuamente y se realiza un monitoreo

constante de las atenciones solicitadas. La solicitud es atendida según la necesidad por instituciones asociadas como la Policía Nacional, Cuerpo de Bomberos, Fuerzas Armadas, Agencia Metropolitana de Transito(AMT) y otros. El sistema cuenta con componentes como: video vigilancia, botones de pánico, geo-referenciación.(Gachet & Rosero, 2015)

Otro sistema de atención a emergencias está relacionado con la seguridad de mujeres, niños y adolescentes a continuación se explican los detalles de la misma.

### **1.3.2. Amiga, Ya No Estás Sola (AYNES)**

Este es un plan estratégico destinado a ofrecer apoyo a las mujeres, niños y adolescentes en general, ante actos de violencia que perjudiquen su dignidad e integridad. Se encuentra implementado en la ciudad de Guayaquil.

Para combatir la violencia se estableció varias alianzas estratégicas con instituciones públicas y privadas dentro y fuera del país, mediante estos convenios se propone minimizar la violencia y brindar apoyo a la mujer. El presupuesto para los recursos necesarios fue adquirido del municipio. El sistema es controlado y monitoreado por especialistas los mismos que, receptan, coordinan y derivan los casos reportados(EL UNIVERSO, 2020).

Asociado a esto, existen campañas a través de los medios de comunicación donde se da a conocer los métodos de apoyo como el botón de pánico, la línea telefónica gratuita, asesoría jurídica y psicológica, casa de protección, becas estudiantiles, talleres formativos y clínicas móviles. (Passailaigue, Galeno, González, & Lozano, 2019)

El botón de pánico es implementado en el presente año 2020, este dispositivo es destinado para las mujeres que sufren agresión verbal o física, se le entrega a la

usuaria después de cumplir ciertos requisitos se realiza una entrevista psicológica y una evaluación de riesgo además de esto debe presentar la boleta de auxilio, en el caso de no poseer una, se presta el servicio para obtenerlo. El dispositivo no puede ser entregado sin una previa capacitación de uso.(EL UNIVERSO, 2020).



En la figura 1, se observa el proceso que se efectúa en un llamado de emergencia, al presionar el botón se activa una alerta con la localización de la usuaria y es enviada a las operadoras mientras que en las oficinas de monitoreo muestra un aviso visual y sonoro que permite tener una reacción rápida a el llamado, el operador designa a un personal acorde al perfil que presente el llamado de auxilio, este proceso se lo realiza de manera rápida. El botón de pánico es monitoreado las 24 horas.(Televistazo, 2020)



El usuario instala en su dispositivo móvil y le da uso a una aplicación, cuando es apropiada y ajustada a las necesidades del usuario por lo que es importante entender los conceptos básicos.

#### **1.4. Aplicaciones Móviles**

Una aplicación móvil, es un programa con diseño y lógica específica para desarrollar una actividad determinada, esta aplicación puede tener finalidades comerciales, administrativos, de protección de datos y seguridad, este software está dirigido para dispositivos móviles. Esta tiene como característica fundamental el libre acceso a los datos en todo tiempo y lugar.(Rodríguez Cerna, 2018)

Todas las aplicaciones independientemente de la finalidad que tengan, necesitan un sistema operativo donde instalarse.

##### **1.4.1. Sistema operativo móvil**

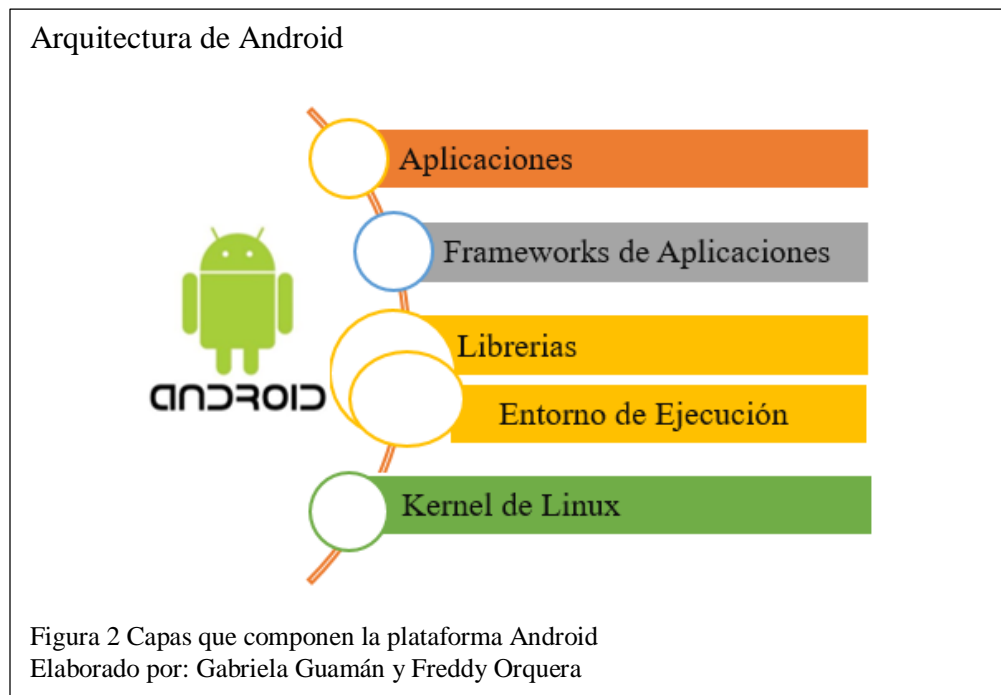
Un sistema operativo móvil es una capa ininteligible encargada de todo el funcionamiento del software en un dispositivo móvil, pues gestiona todos los procesos del sistema permitiendo de este modo la comunicación entre el dispositivo y el usuario.(Silva, 2015)

Los sistemas operativos más populares son:

##### **Android**

Este es un sistema operativo con un diseño y distribución predestinada para teléfonos inteligentes, su plataforma se basa en Linux y actualmente es libre. Está compuesta por un conjunto de software que forman una estructura de capas que interconectan con la aplicación base, creando una pila de software. La plataforma contiene SDK (Software Developer Kit) que suministra herramientas y la API (Application Programming Interface) que son consumidas por distintos softwares.

Las aplicaciones Android utilizan java como lenguaje de desarrollo. (Domínguez Mateos, Paredes Velasco, & Santacruz Valencia, 2014)



A continuación, la figura 2 detalla las capas de la plataforma Android.

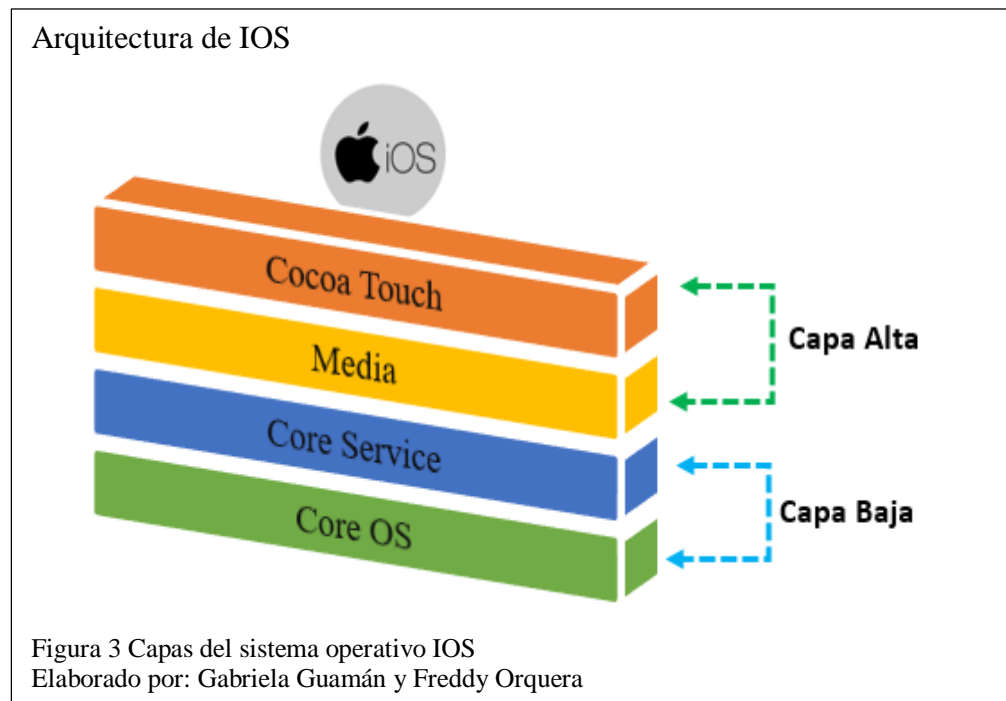
- **Aplicaciones:** Las aplicaciones incluyen diversos aplicativos como correo electrónico, calendarios, contactos, juegos, widgets, etc. Todas las aplicaciones se encuentran desarrolladas en Java.
- **Frameworks de Aplicaciones:** El marco de trabajo se encuentra diseñado para esquematizar la reutilización de componentes, además la APIs del framework posee un acceso abierto para las aplicaciones base.
- **Librerías:** Conjunto de librerías de C/C++ que son utilizadas por los componentes del sistema. Algunas de ellas son: Librería C, 3D, SQLite y otros.
- **Entorno de Ejecución:** Se encuentra ubicado en el mismo nivel de las librerías, este entorno contiene funcionalidades básicas de Java, la ejecución en Dalvik (máquina virtual), es individual para cada proceso y se puede ejecutar varios procesos al mismo tiempo con un uso mínimo de memoria.

- **Kernel de Linux:** Linux es el núcleo de Android donde se brinda servicios de seguridad, red y gestión de procesos y memoria. Esta capa permite la abstracción de los componentes hardware y el stack del software.

## IOS

Sistema operativo fundado por Apple Inc. y con un continuo desarrollo, su código no es abierto por lo que brinda seguridad a los usuarios. Secure Enclave es un co-procesador integrado en un chip SoC (System on a Chip) que maneja la encriptación de claves que ingresan al sistema. Posee una ROM de arranque inalterable que edifica la confianza del hardware.(Apple Inc., 2019)

Posee dos capas principales las altas donde se encuentran los servicios y tecnologías necesarias para el desarrollo y las bajas donde se controla el servicio básico.(Domínguez Mateos et al., 2014)



En la figura 3, se muestra los niveles que intervienen en la arquitectura del sistema operativo IOS, las cuales son:

- **Cocoa Touch:** La capa touch o también llamada táctil contiene una estructura y soporte necesario para la ejecución de aplicaciones.
- **Media:** Es la capa que contiene una tecnología gráfica y multimedia diseñada para brinda una mejor comunicación visual a la aplicación que se muestre en la capa superior.
- **Core Services:** Es la capa que provee el acceso a todos los servicios que utilizan las aplicaciones.
- **Core OS:** Es el núcleo del sistema operativo donde se encuentran los controladores del dispositivo, archivos y entre otros.

Una característica en el diseño técnico que beneficia a la aplicación para marcar una diferencia es la calidad de conexión inalámbrica que posee durante la utilización del aplicativo en el dispositivo, por lo que la arquitectura cliente – servidor debe ser correctamente definida.

#### **1.4.2. Arquitectura de Conexión Cliente – Servidor**

Las aplicaciones móviles constan de 2 elementos principales que intervienen para formar una arquitectura de conexión que nos permita realizar tareas de manera remota.

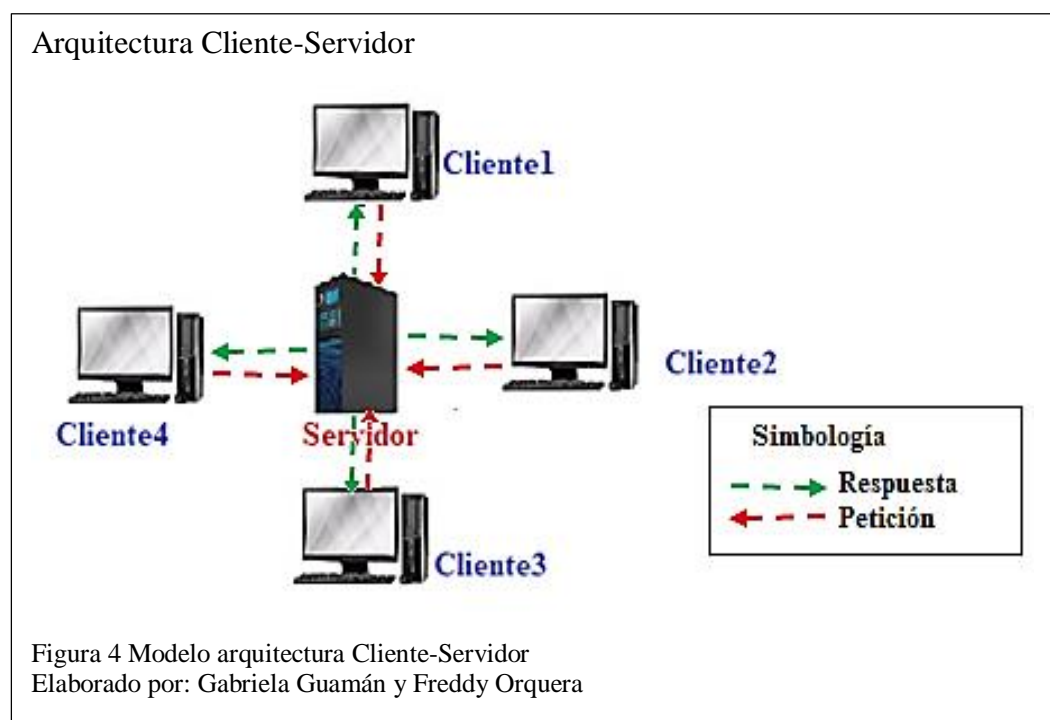
- **Clientes:** Los clientes son los que consumen el servicio mediante solicitudes, estas pueden ser a través de navegadores web o diferentes aplicaciones que nos brinden un acceso.(Vivar Beltrán & Quishpe Sulca, 2019)
- **Servidores:** El servidor hace referencia a un software o hardware que se encarga de atender múltiples solicitudes de clientes proporcionando un feedback. (Felici, 2019)

Cada aplicación posee características únicas dependiendo de las exigencias del usuario, por ello las arquitecturas se han convertido en un tema de gran

importancia al momento de diseñar la red de conexión. Existen diferentes tipos de arquitecturas, estas pueden ser implementadas en las aplicaciones distribuidas entre las más utilizadas se encuentran:

#### 1.4.2.1. *Cliente-Servidor*

Esta es una arquitectura donde el patrón principal es repartir recursos o servicios, es decir, existen proveedores que toman el nombre de servidores y consumidores llamados clientes. En la figura 4 muestra la distribución de trabajo, es decir el servidor se encarga de estar a la escucha de peticiones mientras que el cliente es el que envía la petición. El servidor es capaz de tener una comunicación con varios clientes sin afectar la respuesta, esto quiere decir que los clientes pueden ser n equipos y los recursos solicitados no tendrán un formato predefinido.

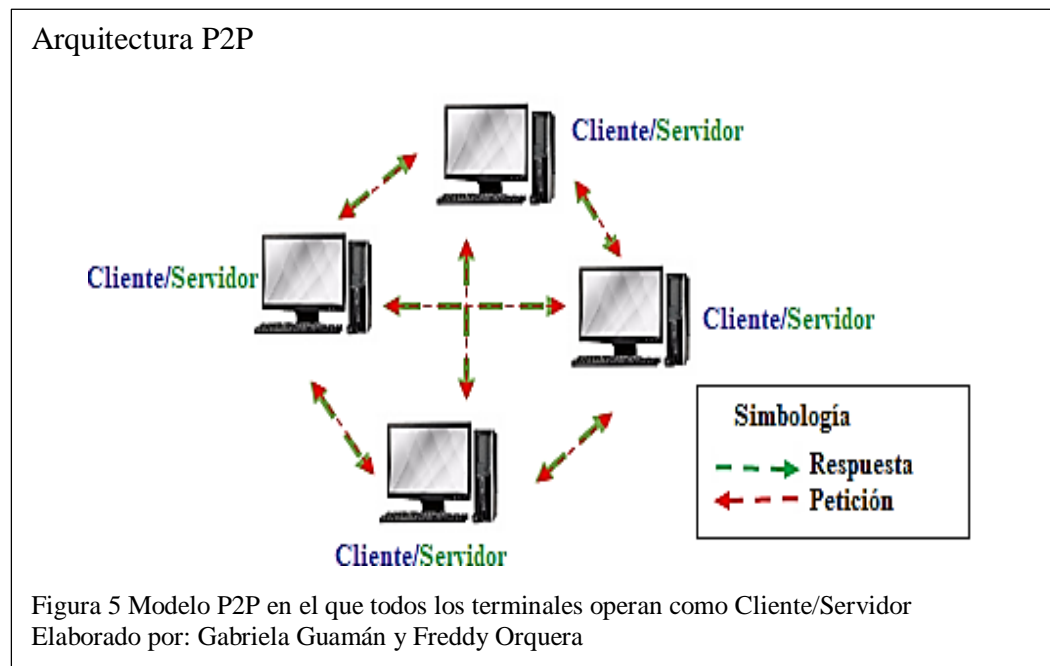


#### 1.4.2.2. *P2P (peer-to-peer)*

Es una arquitectura de red descentralizada en la que los clientes y servidores funcionan de manera equitativa realizando las mismas operaciones, el intercambio de la información se lo hace de forma directa a través de los computadores

interconectados como se observa en la figura 5, el servidor y el cliente pueden solicitar y enviar información entre sí, por lo que no se necesita un servidor centralizado que atienda las peticiones. El cliente podrá solicitar directamente a otro cliente sin necesidad de pasar por un servidor.

Dicha arquitectura brinda un alto rendimiento en conexión y transferencia debido a que la información se comparte entre todos no se requiere únicamente de un servidor. (Jesús Ruiz Calderón, Armando González Cañas & Barrón Pérez, 2014)



Las aplicaciones móviles han ido evolucionando, por lo que es imprescindible conocer la ventaja de la reusabilidad de código y versatilidad que ofrece el desarrollo híbrido.

### 1.5. Aplicaciones híbridas

Las aplicaciones híbridas son desarrolladas con la finalidad de poder ser ejecutadas en distintas plataformas móviles, logrando una adaptabilidad tanto visual como funcional, los sistemas más comunes son Android, iOS. De esta manera, se obtiene una única aplicación para diferentes dispositivos lo que permite utilizar menos recursos sin reducir la seguridad del usuario. (Imlahi Rivas, 2017)

Este tipo de aplicaciones son utilizadas para diferentes propósitos, de igual forma se debe tener en consideración que no se necesita grandes lapsos de tiempo para realizar mantenimientos o actualizaciones debido a su gran ventaja de reutilización de código. A su vez genera un menor costo y una mayor expansión comercial. (Nahuel Delia, 2017)

### 1.5.1. Lenguaje Híbrido

Los lenguajes híbridos mezclan lenguajes de altos niveles creando aplicaciones capaces de tener un rendimiento equitativo en todas las plataformas. Existen varios tipos de lenguajes híbridos uno de los más conocidos es Kotlin.

#### Kotlin

Es un reciente lenguaje de programación con código abierto y de un alto nivel, posee una combinación entre funcional y técnica, brindando una breve adaptabilidad con el código. Kotlin posee la característica de compilar inclusive con versiones antiguas, haciéndolo de un modo rápido y sin inconvenientes. (Kotlin, n.d.)

Este lenguaje puede ser implementado para la creación de nuevas aplicaciones Android, como muestra la Tabla 1, este lenguaje cuenta con varias características que lo destacan y por consiguiente atraen nuevos usuarios sin importar que es nuevo en el mercado. (EBook, n.d.)

Tabla 1 Características KOTLIN

Características	Detalle
<b>Compatibilidad</b>	Cualquier dispositivos Android incluso versiones antiguas.
<b>Rendimiento</b>	Más rápido con respecto a Java
<b>Interoperabilidad</b>	100%
<b>Ejecución</b>	Compacta

<b>Compilación</b>	Rápida y eficiente
<b>Aprendizaje</b>	Adaptable

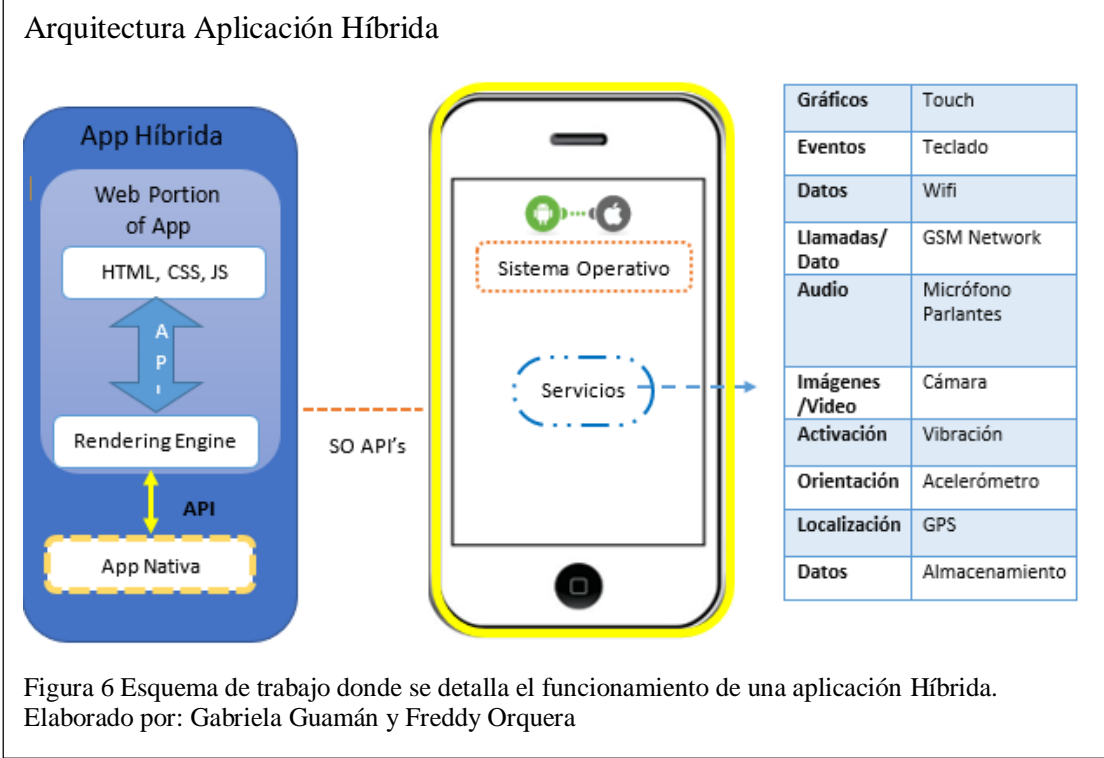
Nota: Las características relevantes de KOTLIN.

#### ***1.5.1.1. Características de los lenguajes híbridos***

Las aplicaciones híbridas sacan el mayor provecho en el desarrollo para luego ser utilizadas por los usuarios, debido a que son amigables y fáciles de manipular. Las aplicaciones en sus principios fueron destinadas con la finalidad de ser usadas en una sola plataforma y en un determinado dispositivo, sin embargo, la tecnología ha ido avanzando y se ha ido extendiendo, por lo que se ha implementado en diversos dispositivos tales como televisores, relojes inteligentes entre otros.(Fundació Jesuïtes Educació, 2019)

En la figura 6 se puede observar una representación de la arquitectura híbrida donde se desarrolla un núcleo común con tecnologías web como HTML, CSS, JavaScript continuamente se recubre con un contenedor específico para cada uno, por lo que permite automatizar generando un proceso multiplataforma donde permite realizar una vez y distribuir la aplicación a múltiples plataformas como iOS, Android, Windows pone, entre otras. Para una mejor experiencia de usuario se puede emular las interfaces debido a que permite tener acceso ilimitado al hardware del dispositivo como cámaras, audio, GPS y otros. Todo esto ofrece responder de una forma rápida a las necesidades del usuario y del mercado.





### 1.5.2. Frameworks

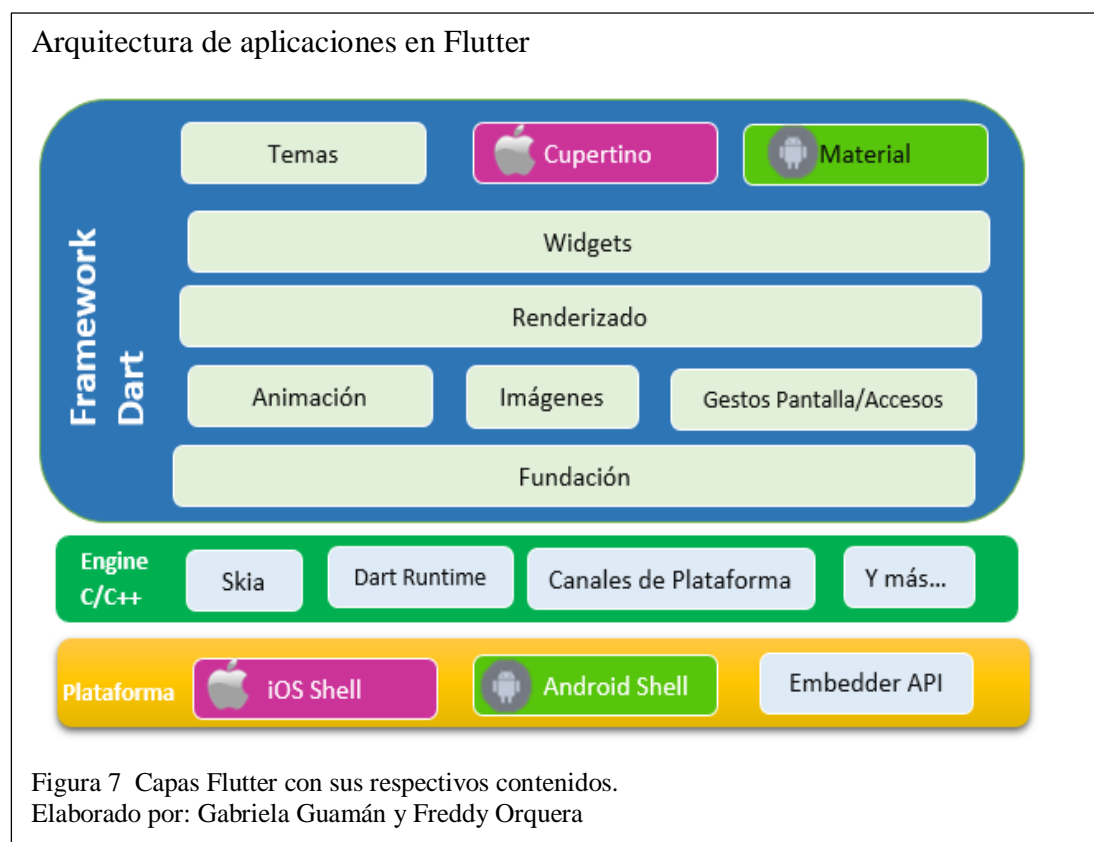
Es considerado como un marco de trabajo adaptable en la que se puede añadir varios complementos presentando una estructura o patrón para el desarrollo e implementación de una aplicación definida promoviendo las buenas prácticas. Este puede incluir bibliotecas, scripts entre otros softwares que brinden apoyo para el desarrollo o unión de componentes. (Tulcán Suquitana, 2020)

En el prototipo de la aplicación móvil multiplataforma propuesta se contó con los siguientes Frameworks:

#### Flutter

Es un framework que brinda una serie de beneficios y características para el desarrollo de una aplicación de alto rendimiento, se adapta a diferentes plataformas y es de código abierto destinado para diseñadores, prototipadores y programadores, ofreciendo una experiencia agradable e interesante. (Hernandez, 2019)

Este contiene 3 pilares fundamentales; la velocidad es una de las principales particularidades debido a que se puede visualizar las actualizaciones de manera inmediata sin tener que interrumpir la ejecución, otro aspecto importante es el diseño que ofrece pues, contiene animaciones y efectos personalizables brindando una apariencia agradable al usuario, por último, se menciona la tenencia de comportamientos propios de cada componente sea en iOS o Android, estas pueden ser navegaciones entre pantallas, botones , etc.(Vidal, Vilalta, 2019)



Este framework abarca distintas capas que se construyen secuencialmente, en la figura 7 se observa que se encuentra desde la más frecuente a la menos utilizada, la finalidad de esto es ayudar a reducir el código. La capa superior contiene librerías, material donde se brinda la posibilidad de usar widgets predeterminados, creados o combinados, para esta última opción es necesario acudir a capas inferiores, la segunda capa contiene las bibliotecas escritas en C/C++ y finalmente

en la primera capa se especifica la plataforma y se configura el renderizado.  
(Angulo, 2018)

### 1.5.3. Ventajas y desventajas

Actualmente las aplicaciones nativas como híbridas son publicadas y utilizadas por el usuario incluso muchas de ellas son instaladas por defecto y en otras es necesario buscarlas en tiendas digitales. Las aplicaciones independientemente de su finalidad deben ser desarrolladas con el propósito de brindar facilidad y flexibilidad, permitiendo de esta manera un funcionamiento sin fallos y disponible para el usuario. No obstante, a pesar de todos los beneficios que presume existen ciertas inconformidades, la Tabla.2 muestra ciertas ventajas y desventajas que implican la producción de una aplicación híbrida.

Tabla 2 Ventajas y desventajas de aplicaciones híbridas

APLICACIONES HÍBRIDAS	Ventajas	Desventajas
	Código único	Lenguaje con altos conocimientos.
	Reutilización de código	No tiene acceso a todos los recursos
	Menor costo e inversión	Rendimiento bajo debido a la renderización con el servidor.
	No requiere permisos externos de publicación en tiendas digitales	Acceso no libre en algunas API's.
	Actualización rápida	La velocidad puede verse afectada.
	Menor costo en mantenimiento	Interfaz no atractiva.
	Interfaz adaptable e intuitiva	
	Prototipado rápido	

Nota: Información relevante de ventajas e inconvenientes de las aplicaciones híbridas.

Para un mejor diseño de una aplicación es importante saber cuál será su finalidad, si es conveniente crearlo de manera híbrida teniendo en cuenta que existen ciertas

limitaciones de recursos en dispositivos móviles. El desarrollo multiplataforma cuenta con mayores beneficios principalmente en tiempo y costo, además de la adaptabilidad de interfaces, pues, el diseño es un aspecto importante pues el usuario va a interactuar con la aplicación y necesita una conexión simple y rápida. La reutilización de código presenta una ventaja para los desarrolladores debido a que se necesitan actualizaciones para mantener una aplicación efectiva y sin inconvenientes, teniendo como propósito lograr una experiencia acogedora para promover una fidelidad del usuario con la aplicación.

## **1.6. SCRUM**

SCRUM es un marco de trabajo definido como proceso ágil donde se gestiona y controla el desarrollo del software de manera iterativa e incremental. Al ser adaptativo y transparente en su progreso brinda una eficacia relevante en la gestión y desarrollo del proyecto.

Esta metodología tiene un desarrollo simple se basa en iteraciones o también llamadas Sprint, las cuales son establecidas de una a cuatro semanas dependiendo la necesidad. El resultado de cada iteración influirá en el proceso, pues al depender una de otra afecta el tiempo de entrega del producto final. Sin embargo, en ocasiones también intervienen cambios en los requisitos por parte del usuario, por lo cual la flexibilidad e innovación son aspectos fundamentales para asimilar estas alteraciones.(Onieva López, 2018)

Para tener una idea más clara se detallará algunos de los aspectos y componentes relevantes de esta metodología.

### **1.6.1. Características**

Scrum se aplica y se adapta en problemas complejos al ser liviano y fácil de entender no se necesita tener una dominación completa del tema, cabe destacar que esta metodología no es un proceso o técnica. Esta metodología utiliza tres

pilares fundamentales para el control de los procesos: transparencia donde los responsables y colaboradores pueden observar el trabajo de manera clara y concisa, otro de los pilares es la inspección de los progresos y herramientas para detectar desviaciones no proyectadas y finalmente la adaptación donde se ajusta el proceso para disminuir el impacto del cambio.(Payano Muñoz, 2019)

### 1.6.2. Ventajas y Desventajas

La metodología posee ventajas e inconvenientes que se presentan y se debe tener en cuenta para su aplicación.

Tabla 3 Ventajas y Desventajas de Scrum

Ventajas	Desventajas
<b>Desarrollo ágil</b>	Tareas concluidas de manera inadecuada.
<b>Flexibilidad y adaptación a cambios</b>	Fechas de entregas apresuradas.
<b>Tareas Equitativas</b>	Estrés
<b>Resultado a corto plazo</b>	Equipos no siempre auto-organizados

Nota: Tabla donde se puede apreciar los beneficios e inconvenientes que presenta la metodología Scrum.

### 1.6.3. Componentes

Los componentes que intervienen en el desarrollo del proyecto en la metodología Scrum son (Siqueira Simoes, 2019):

#### Roles

- **Dueño del producto (Product Owner):** Persona responsable del producto que se va a brindar al cliente, es el que enlista los objetivos de trabajo y necesidades para posteriormente ser desarrolladas.

- **Equipo Scrum:** Grupo de trabajo que se encarga de organizar, crear y desarrollar lo necesario para entregar un producto terminado, también, debe tener un trabajo organizado y colaborativo.
- **Scrum Master:** Responsable de hacer comprender y aplicar la metodología de trabajo tanto en teoría como en práctica, respetando siempre las pautas establecidas del mismo modo se encargará de hacer entender al personal externo la necesidad de su presencia en algunas interacciones.

### Eventos

- **Sprint:** El corazón de Scrum, donde avanza de modo incremental con un tiempo máximo de un mes en el cual se mantiene un objetivo y un equipo de trabajo que ejecuta las tareas necesarias para completar y poder pasar al siguiente sprint.
- **Sprint Planning:** La planificación de cada sprint de manera colaborativa con todo el grupo de trabajo respetando las horas destinadas al trabajo y los resultados que se deben entregar.
- **Daily Scrum:** Reunión diaria donde se realiza una retroalimentación de la iteración de cada miembro del equipo donde es necesario preguntarse: ¿Qué se hizo ayer?, ¿Qué se hará hoy?, ¿Existe algún inconveniente?
- **Sprint Review:** Reunión que se ejecuta en cada entrega de avances donde se comprueban las funcionalidades establecidas.
- **Sprint Retrospective:** Reunión informal que se concentran en la calidad de trabajo y establece nuevas modalidades que ayudan a obtener un mejor desempeño.

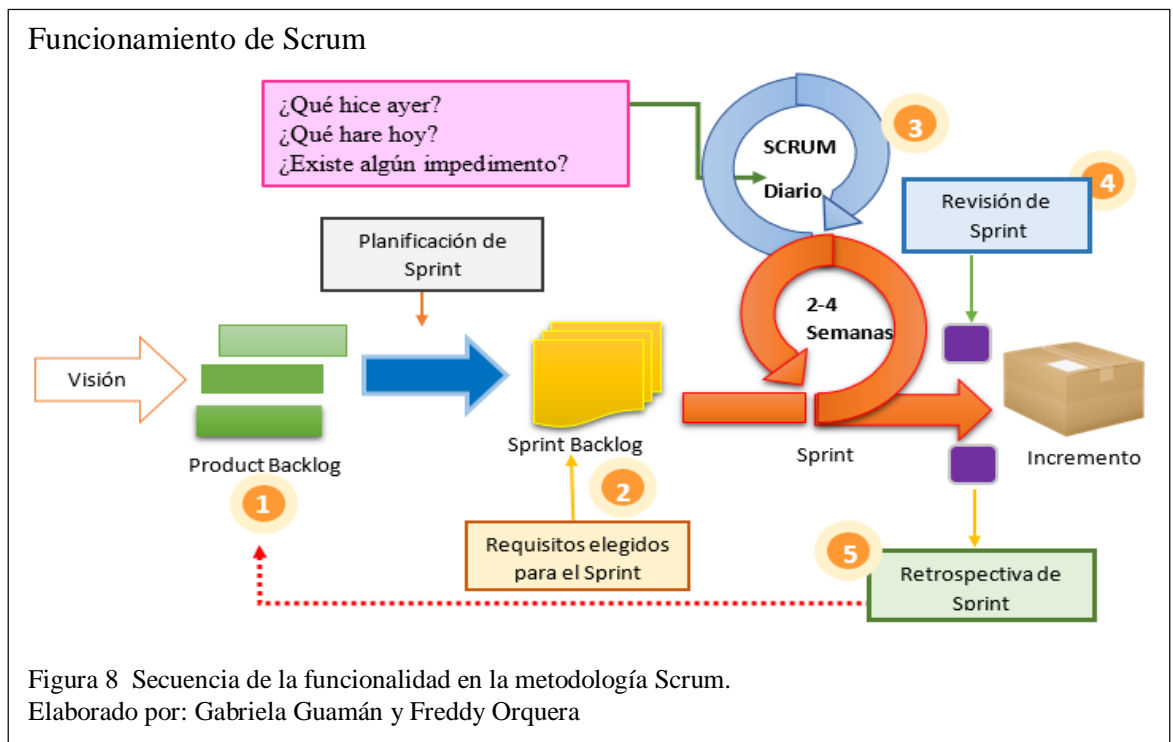
## Artefactos

- **Product Backlog:** Lista de requerimientos necesarios para la entrega completa del producto.
- **Sprint Backlog:** Lista de tareas que se ejecutan durante el Sprint para generar un incremento satisfactorio.
- **Product Increment:** Resultado de los Sprint finalizados con éxito.

### 1.6.4. Funcionamiento

Para una mejor comprensión del ciclo de Scrum es necesario conocer cómo funciona y que componentes intervienen en el mismo.

Scrum al considerarse una metodología ágil es muy utilizada en el desarrollo de software, en la figura 8 se observa las fases que se aplican en el modelo antes mencionado



## **Fase 1: Definir – Product Backlog**

En esta fase siendo la primera, es necesario tener una visión y elaborar un Backlog del Producto o también llamada Pila de Producto, este será el principal requisito para poder empezar. Se realiza una lista priorizando los requerimientos o funcionalidades solicitadas por el cliente, a estas tareas se les denomina historias de usuario, donde se permite tener una identificación y manejo adecuado de cada trabajo. (Malpica Velasquez, 2014)

Usualmente cuenta con los siguientes campos:

- **Número:** Identificador auto- incremental y único, por lo que no se admite un registro duplicado permitiendo de esta manera no perder el rastro de una historia, aunque exista cambio de nombre.
- **Nombre de historia:** Descripción corta que brinda una idea clara al cliente del proceso que se realiza, debe ser distinta y generalmente contiene un máximo de 10 palabras.
- **Importancia:** Escala de importancia que se le da a una historia, el rango será especificado por preferencia del usuario donde puede ser numérica, categórica, intervalo, entre otros. Se recomienda no definirlo como prioridad, pues resulta incómodo decir que una historia es más importante que otra.
- **Puntos estimados:** Estimación del tiempo que dura una historia en ser concluida, esto incluye realización de pruebas respectivas que permitan brindar una garantía para ser integradas en el proyecto acumulativo.
- **Descripción:** Detalle del comportamiento del módulo, demostrando las funcionalidades que cumplen y que resultados se deben obtener.
- **Observación:** Información adicional que servirá para un análisis posterior, o simplemente un recordatorio de alguna observación necesaria.



Tabla 4 Ejemplo de Pila de Producto

Pila de Producto					
ID	Nombre	Imp.	Pts. Estimados	Descripción	Observación
1	Registro de usuario	Alta	4	Registrar la información requerida para posteriormente iniciar sesión.	Ingreso con datos validados
2	Perfil de usuario	Alta	2	Observar en la pantalla de perfil los datos ingresados en el registro.	Demostrado

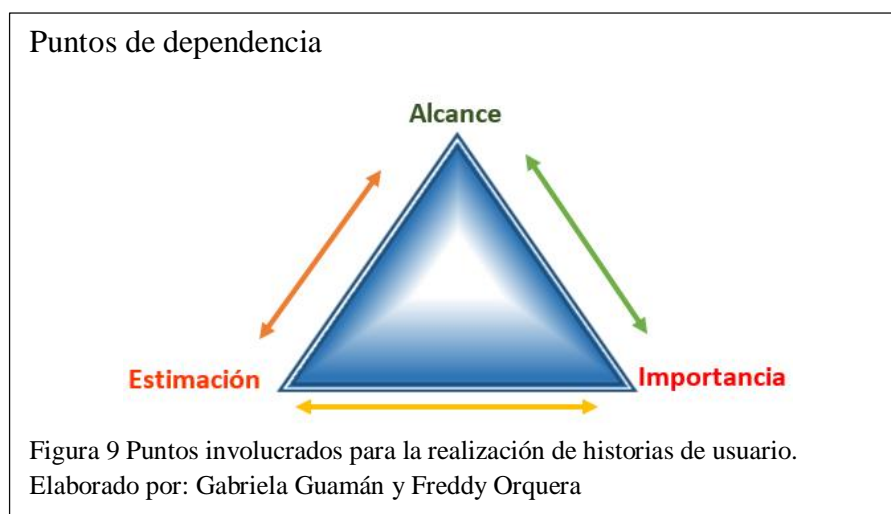
Nota: Ejemplo donde muestra una pila de producto en la que se listan historias registradas en cada sprint.

La Tabla. 4 se la puede realizar en cualquier software siempre y cuando sea trabajo cooperativo, pues los sprint irán variando de historia como de responsable por lo que se necesita una simultanea hoja de trabajo donde se podrá mostrar a todos los integrantes el avance del proyecto. En la pila de producto se puede aumentar campos adicionales que brinden un seguimiento o control de una forma más precisa y detallada.

### Fase 2: Planificar– Sprint Backlog

En esta fase es importante realizar una revisión rápida de la pila de producto, donde se verificará que todos los campos estén correctamente definidos y no exista inconvenientes. Otro aspecto significativo es el aumento de historias por parte del cliente, en este caso se replanteará la pila y se ajustará adecuadamente al cambio para adaptar la planificación del Sprint y ejecutarlo de manera correcta y sin inconvenientes.

Es importante que el cliente sea participe de la planificación de Sprint, pues es indispensable la aprobación de cada módulo y revisión de avances, aunque en muchos de los casos el cliente no se encuentra disponible 100% para estas reuniones. Es indispensable que el equipo de trabajo y el dueño del producto se encuentren presentes en la planificación del Sprint, debido a que intervienen 3 puntos dependientes uno de otro que son principales en cada historia como se observa en la Figura 10.(Kniberg, 2007)



En la figura 9 se observa la importancia del cliente, pues es el encargado de colocar las metas de cada Sprint y darle la importancia a cada historia mientras que el equipo de trabajo lo revisa y se encarga de estimar el tiempo para cada historia. Conforme se avance el proyecto, surgirán cambios de estimaciones debido a que el usuario puede llegar a solicitar con anticipación lo que obliga a replantear el objetivo de dicha historia y el alcance que tendrá pues el tiempo será re-estimado para lograr un acuerdo que ofrezca una objetivo y resultado adaptado al tiempo que se reajuste.

En el triángulo formado por los puntos de dependencia es necesario tener en cuenta el tema de calidad, donde se convierte en un punto adicional indispensable. Este se divide en interna, el cual está dirigido a los usuarios y externa, la que no es

visible para el usuario, pero es una parte importante en el mantenimiento del sistema. Otro aspecto importante es la reunión de planificación para cada Sprint, ya que al tener una lista de puntos a tratar se podrá crear un ambiente de orden y no permanecer un tiempo excesivo en un solo punto, también es recomendable crear un documento de constancia del acuerdo final de cada reunión para tener un respaldo ya sea de beneficio para el cliente o para el grupo de trabajo. (Malpica Velasquez, 2014)

### **Fase 3: Reunir – Scrum Diario**

En la tercera fase está pensada en la planificación de las reuniones sean virtuales o presenciales. La finalidad de esta es definir cómo se va a empezar a trabajar, por tanto, es importante tener en cuenta las ventajas y desventajas que puedan presentar a los miembros del equipo al momento de seleccionar una hora o lugar.

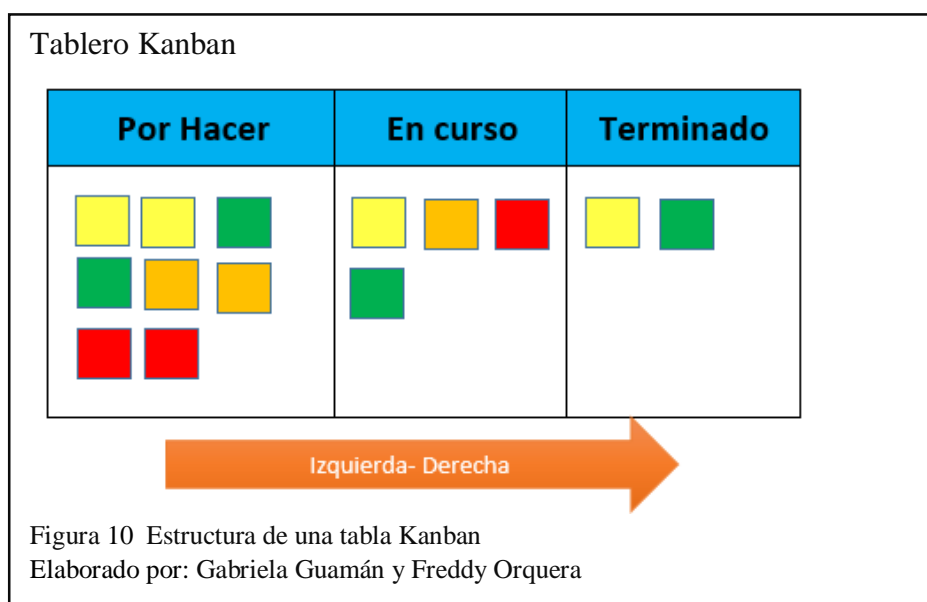
Para esta reunión se debe tener actualizada la información de la pila, ya que se comunicará el progreso realizado por cada miembro del equipo, con la finalidad de observar el avance del trabajo y exponer dudas o inconvenientes que surgen de manera imprevista, para lo cual se establecen medidas de resolución ante el problema y de esta forma evitar retrasos de largos lapsos de tiempo. Esta reunión no durará más de 15 minutos y se obtendrá como resultado final una información actualizada en la pila de Sprint y el trabajo que se realizara hasta la siguiente reunión.(Francia, 2017)

Para el equipo de trabajo es importante poder observar y conocer de manera rápida el progreso de la pila en cualquier momento que se lo desee, debido a esto dependiendo del grupo de trabajo se establecen técnicas necesarias que se adapten a lo requerido.

## Kanban

Un sistema que brinda una representación visual continua del avance diario del Sprint. Es necesario buscar un espacio libre en una pared o lugar plano que sea visible para todos los integrantes del equipo de trabajo la medida de este espacio dependerá de la cantidad de personas en el equipo la medida sugerida es de 2x2 metros. El tablero ayudara a identificar cuellos de botella existentes en algún proceso.

El esquema con el que se trabaja este compuesto por 3 campos: Por hacer, en curso y terminado como se representa en siguiente imagen.



En la figura 10 se observa la estructura del tablero donde el equipo de trabajo iniciará sus actividades para lo cual se establecerá un número de límite de trabajo simultaneo en la columna “En curso”. A continuación, se detallan los pasos a seguir para un correcto manejo del tablero.

El equipo identificara las columnas establecidas en el tablero y el orden en el que se colocaran las tareas, este será de izquierda a derecha. Una vez definido el límite de trabajo se lo colocara en la columna correspondiente con las siglas WIP (work

in progress) y el número acordado de modo que se observara de la siguiente manera **WIP=2**. Todos los miembros del equipo pueden visualizar el proceso de cada tarea evitando aglomeraciones a una sola persona, ya que permitirá replantear tareas si fuese necesario.(Palacio, 2020)

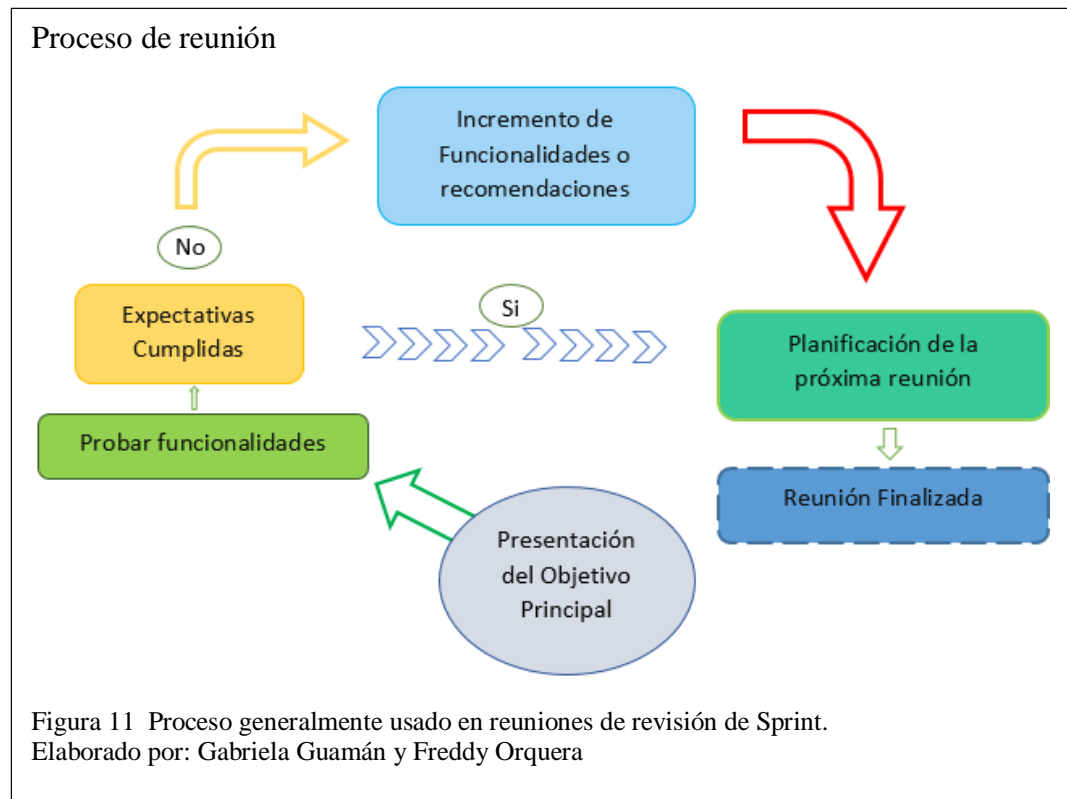
Para poder identificar que no exista aglomeraciones de tareas es indispensable etiquetar cada tarea con el nombre de la persona que ha sido designada, también se puede observar tareas que no avanzan o se encuentran estancadas permitiendo de esta manera colocar como tema de importancia en la reunión diaria, pues se debe establecer una solución inmediata para no tener retrasos. Si se maneja correctamente la tabla se tendrá como efecto una mejora en el flujo de trabajo y buenos resultados.

#### **Fase 4: Revisar – Revisión de Sprint.**

Esta fase es primordial para el incremento del proyecto, se realiza una reunión donde se entregan las actividades ya terminadas y se presentan los inconvenientes que se presentaron durante el desarrollo. En la revisión se encuentran presentes el equipo de trabajo, Scrum Master, Power Owner, la reunión tiene una duración máxima de cuatro horas donde se presentan las funciones implementadas en el Sprint. El modulo que se revisa debe ser correctamente probado si es posible se lo presenta en un servidor de pruebas donde se asimile de mejor manera el funcionamiento al presentarlo ante el cliente. No se puede mostrar un trabajo inconcluso, pues al ser expuesto con fallas o funcionalidades incompletas crea una confusión al usuario. (Menzinsky, López, & Palacio, 2019)

En la figura 11 se observa el proceso generalmente utilizado en la revisión de Sprint, donde se presenta el objetivo principal a los miembros de la reunión que fueron elegidos para asistir a la revisión, seguidamente se comprueba el correcto

funcionamiento y cumplimiento de las expectativas del usuario dependiendo de la aceptación puede ir directo a la planificación de la próxima reunión o al incremento de funcionalidades o recomendaciones que necesite el sprint, finalmente, la reunión será concluida.



### Fase 5: Regresar –Retrospectiva de Sprint

En esta última fase es imprescindible establecer un lugar y una hora que permita asistir a todos los miembros del equipo debido a que este punto no siempre es aceptado en el grupo de trabajo, sin embargo, es una buena práctica para percibir nuevas sugerencias, ideas y contribuciones para evitar errores en el próximo Sprint. La reunión no es necesariamente una reunión formal, pues es necesario sentirse cómodo para entablar una discusión que puede variar de 1 a 3 horas dependiendo las inquietudes que aparezcan, esta no debe ser interrumpida. (Trigas Gallego, 2012)

Una vez iniciada la conversación cada miembro del equipo que se encuentra presente tendrá la oportunidad de expresar su punto de vista sea positivo o negativo sin interrupciones pues cada uno tiene una forma libre de pensar o dificultades que a los demás no le parecen importantes. Para tener una mejor visión se utiliza una pizarra donde se organizan las mejoras propuestas, esta se divide en tres columnas:

- **Bueno:** Se lo repite nuevamente en el siguiente Sprint.
- **Mejorable:** Realizar de forma diferente para evitar contrariedades.
- **Mejoras:** Cambios para optimizar el próximo trabajo.

Tomando en cuenta las columnas y la importancia de cada idea es preciso escoger un número máximo de cambios que sean concretos y reales. Es recomendable no escoger numerosos cambios ya que al aglomerarse de mejoras podría crear un conflicto en el siguiente sprint. (Malpica Velasquez, 2014)

Cada grupo de trabajo debe establecer sus condiciones de retrospectiva donde se aprecie la participación de todo el personal que interviene en el desarrollo del proyecto.

## **Capítulo 2**

### **2.1. Análisis de Factibilidad**

Este punto tiene como objetivo la comprensión del análisis y diseño del prototipo, para esto se comenzará con el análisis de factibilidad donde los requerimientos serán un punto importante para proseguir con el diseño y finalmente el desarrollo e implementación del prototipo.

#### **2.1.1. Factibilidad Técnica**

La factibilidad técnica estudia la viabilidad del hardware y software necesarios para una correcta elaboración del proyecto, en este caso la aplicación hace uso de herramientas libres y confiables que al ser implementadas de manera adecuada se tendrá como resultado una aplicación multiplataforma.

Para el desarrollo en Android no es necesariamente tener un conocimiento avanzado, pues dependiendo del progreso se necesitará auto educación, aunque al conocer otros tipos de lenguaje resulta fácil adaptarse. En cuanto a la creación y manejo de la base de datos el conocimiento es básico.

A continuación, se detallan los requerimientos de hardware y software indispensables para el proceso de desarrollo y ejecución del proyecto.

#### **Especificaciones mínimas de Hardware**

En la Tabla 5 se puede apreciar los requerimientos mínimos para el desarrollo, pruebas respectivas y ejecución apropiada de la aplicación. Las características fueron identificadas al momento de instalar, desarrollar y ejecutar la aplicación.



Tabla 5 Requerimientos de Hardware

Características	Requisitos Mínimos
<b>Procesador</b>	Intel i3 o superior de 3.4 GHz
<b>Sistema operativo</b>	Mac OS / Windows
<b>Almacenamiento</b>	480 GB de Disco Duro
<b>Memoria Ram</b>	8Gb
<b>Red</b>	Conexión a Internet
<b>Dispositivo</b>	Computador portátil /mesa

Nota: Las características necesarias para el desarrollo del prototipo.

### Especificaciones mínimas de Software

En la Tabla 6 se detalla los requerimientos mínimos de software y el tipo de licencia que será utilizado.

Tabla 6 Requerimientos de software.

Software	Tipo de Licencia	Uso
<b>S.O</b>		Sistema Operativo
<b>Android Studio</b>	Gratuita	Editor de Código
<b>Google Chrome</b>	Gratuita	
<b>Gimp</b>	Gratuita	Editor de imágenes
<b>Mendeley 1.19.4</b>	Gratuita	Referencias Documentación
<b>Cloud Firestore</b>	Cuota Gratuita	Almacenamiento de Información
<b>SDK de Android</b>	Gratuita	Librería
<b>Emulador</b>	Gratuita	Pruebas de Funcionamiento
<b>cupertino_icons v0.1.2</b>	Gratuita	Librería (Estilo de iconos )

<b>generic_bloc_provider v1.0.9</b>	Gratuita	Librería (Interfaz nativas)
<b>firebase_auth</b>	Versión Gratuita	Librería (Autenticación )
<b>image_picker v0.6.1+4</b>	Gratuita	Librería (Imágenes con cámara)
<b>cached_network_image v1.1.1</b>	Gratuita	Librería
<b>firebase_storage v3.0.6</b>	Versión Gratuita	Librería (Almacenamiento)
<b>cloud_firestore v0.12.10</b>	Gratuita	Librería (Base de Datos)
<b>geolocator v5.1.4+2</b>	Gratuita	Librería (Ubicación Tiempo real)
<b>google_maps_flutter v0.5.21+11</b>	Versión Gratuita	Librería (Google Maps)

Nota: Tabla de herramientas necesarias para un funcionamiento adecuado.

### Especificaciones mínimas en Recurso Humano

La Tabla 7 muestra la distribución del personal necesario para organizar, gestionar y cumplir las funciones asignadas para el éxito del proyecto. Hay que tomar en cuenta que son 2 personas encargadas del proyecto.

Tabla 7 Recurso Humano

Descripción	Cantidad disponible	Cantidad solicitada
<b>Diseñador de Base de Datos</b>	2	1
<b>Diseñador de Interfaz</b>	2	1
<b>Desarrollador</b>	2	2

Nota: Tabla de recurso humano que intervendrá en todo el proyecto.

## Conclusión

En el análisis de factibilidad técnica se requiere de recursos para la ejecución completa del proyecto, en la parte de software se puede observar que en su mayoría son de adquisición gratuita debido a lo cual no presenta gran dificultad para su obtención, en la parte de hardware son características mínimas y económicas capaces de ser adaptadas al equipo si se lo necesitara por lo que no representa inconveniente y se puede afirmar que el proyecto es factible técnicamente.

### 2.1.2. Factibilidad Económica

La factibilidad económica incluye un análisis de costos y beneficios que intervienen en la ejecución, sin embargo, la finalidad del proyecto está dirigido para prestar ayuda social por lo que no se menciona un análisis de ingreso por la aplicación.

A continuación, se detalla una estimado de costos para software, hardware, insumos y servicios de personal.

Tabla 8 Insumos y Servicios

Detalle	# Personas	Tiempo	Valor Unitario	Valor Total
Internet	2	6	40,00	240,00
Personal de Desarrollo	2	6	750,00	9.000,00
Servicios Básicos	2	6	16,00	96,00
Cursos en Android	2	1	150,00	300,00
Total:				\$ 9.636,00

Nota: En la tabla se especifica los valores de cada recurso utilizado en insumos y servicios.

Tabla 9 Costos Hardware

Detalle	Características	Valor Unitario	Valor Total
<b>Disco Duro Solido</b>	Ssd 2.5 Interno 1TB	120,00	250,00
<b>Teléfonos de prueba</b>	Plataforma Ios y Android	0,00	0,00
<b>Total:</b>			<b>\$ 250,00</b>

Nota: En la tabla se especifica los valores de cada recurso utilizado en hardware.

Tabla 10 Costos Software

Servicio	Detalle	Valor Unitario	Valor Total
<b>Cloud Firestore</b>	1 GB almacenamiento 600,00 escritura 1.500,00 lectura 600,00 eliminación	0,00	0,00
<b>Realtime Database</b>	1 GB almacenamiento 10 GB transferencia	0,00	0,00
<b>Librerías</b>	Versiones Gratuitas	0,00	0,00
<b>Android Studio</b>	Versión 3.6.1	0,00	0,00
<b>Total:</b>			<b>\$ 0,00</b>

Nota: En la tabla se especifica los valores de cada recurso utilizado en software.

Tabla 11 Total en Recursos

Recurso	Valor Total
<b>Insumos y Servicios</b>	\$ 9.636,00
<b>Hardware</b>	\$ 250,00
<b>Software</b>	\$0,00
<b>Total:</b>	<b>\$ 9.886,00</b>

Nota: En la tabla se especifica el valor total de recursos utilizados.

## **Conclusión**

En el análisis económico se ha estimado un valor mensual que corresponde a 8 horas de trabajo y en días laborables, la adquisición del complemento para la mejora del equipo de desarrollo no involucra un gasto inútil pues brindará un mejor rendimiento a los equipos incluso para posteriores trabajos, los teléfonos no tienen ningún valor, debido a que se utilizarán los personales de los dos integrantes obteniendo de esta forma variación de plataformas para las pruebas requeridas. Como se mencionó anteriormente los recursos de software no tienen valor ya que para el prototipo se utilizará solo versiones gratuitas, analizando todos los recursos y tomando en cuenta que el valor total será asumido por los integrantes del proyecto Gabriela Guamán y Freddy Orquera se establece que el proyecto es económicamente factible.

### **2.1.3. Factibilidad Operacional**

A nivel nacional los teléfonos celulares están presentes en el hogar según el Instituto Nacional de Estadística y Censos en el año 2018 el porcentaje creció un 57.9%, al igual que las personas que utilizan el internet aumento un 20.7 %.(INEC, 2018) Ante este escenario, se puede mencionar que el desarrollo de la aplicación móvil al ser adaptada a cualquier dispositivo móvil los ciudadanos podrán manejarlo de una manera rápida, incluso sin capacitación debido a su interfaz amigable e intuitiva lo que facilita su manejo.

Desde el punto de vista operativa, el prototipo se lo presenta como un proyecto innovador el cual obtiene una captación instantánea con el cliente, debido a la gran necesidad de apoyo en la seguridad personal. Otro aspecto que influye es el desarrollo, al ser multiplataforma, las futuras actualizaciones se las realizara de forma rápida y estratégica para una mejora continua en la aplicación. Otra de las

ventajas es que la mayoría de usuarios cuentan con dispositivos inteligentes lo que no representa un gasto adicional para poder instalar la aplicación.

En el análisis también se considera un inconveniente que puede mostrar la aplicación ante la sociedad al no ser bien explicado su función algunos usuarios podría oponerse al uso por invasión a la privacidad, ya que la podría tomar como un rastreador personal donde los contactos puedan saber su ubicación en todo momento sin autorización.

### **Conclusión**

Al evaluar el problema de estudio y la posible solución que ayudará a una comunicación en tiempo real de la emergencia, el estudio de factibilidad realizada muestra que la aplicación al ponerse en marcha será una herramienta usada por los ciudadanos con una gran adaptabilidad y buena experiencia haciendo de este modo un proyecto viable, aunque exista el inconveniente antes mencionado.

## **2.2. Análisis de Requerimientos**

En el análisis de requerimientos se expone los requerimientos que fueron establecidos para el desarrollo del prototipo. En este punto se define perfiles de usuarios, interfaces y funcionalidades operacionales que tendrá la aplicación, verificando siempre las solicitudes planteadas por el cliente.

### **Propósito**

Determinar los servicios que proporciona el sistema mediante los requerimientos funcionales y no funcionales describiendo de esta manera los servicios, funciones, métodos de seguridad, almacenamiento, entre otros. Permitiendo de este modo una revisión y constancia de los registros de cambio o aprobación que ofrece la aplicación.

## Personal involucrado

Tabla 12. Datos del Personal

Referencia	Datos
<b>Nombre:</b>	Gabriela Guamán
<b>Rol:</b>	Equipo scrum
<b>Profesión:</b>	Ingeniería en Sistemas
<b>Responsabilidad</b>	Diseñador , modelador de base de datos y desarrollador
<b>Información de contacto</b>	<a href="mailto:gguamang@est.ups.edu.ec">gguamang@est.ups.edu.ec</a>
<b>Nombre:</b>	Freddy Orquera
<b>Rol:</b>	Equipo scrum
<b>Profesión:</b>	Ingeniería en Sistemas
<b>Responsabilidad</b>	Diseñador , modelador de base de datos y desarrollador
<b>Información de contacto</b>	<a href="mailto:forquera@est.ups.edu.ec">forquera@est.ups.edu.ec</a>
<b>Nombre:</b>	Julio Proaño
<b>Rol:</b>	Product Owner , Scrum Master y Usuario final
<b>Profesión:</b>	Ingeniería en Sistemas
<b>Responsabilidad</b>	Proveer de información necesaria para satisfacer las necesidades del usuario final
<b>Información de contacto</b>	<a href="mailto:jproano@est.ups.edu.ec">jproano@est.ups.edu.ec</a>

Nota: Información del personal involucrado en el proyecto.

### **2.3. Alcance del Prototipo**

Se desarrollará un prototipo de una aplicación móvil que permita enviar un llamado de emergencia a sus contactos registrados. Los usuarios podrán acceder a la aplicación sin restricciones y el registro se lo realizará de manera simple brindando facilidad a los usuarios.

Los contactos serán registrados de una manera ágil incluso con fotografías para una mejor identificación. El servidor permitirá que la aplicación pueda interactuar de manera bidireccional con el usuario generando una respuesta en cualquier lugar, la misma que es realizada de forma invisible para el usuario, todo el código que se ejecutará no será mostrado, únicamente se presentan los resultados esperados.

La aplicación está destinada para multiplataforma por lo que no existirá inconveniente en la instalación y funcionalidad de esta.



## Capítulo 3

Las metodologías ágiles permiten adaptarse al proyecto ofreciendo grandes ventajas que permiten adaptar el proyecto al entorno. Para este proyecto se ha seleccionado la metodología SCRUM.

### 3.1. Implementación de la metodología

Scrum, tiene un desarrollo simple y uno de los puntos principales es el trabajo en equipo, al ser una metodología sin un plan se sostiene a comportamientos variables por lo que depende de cada escenario presentara un comportamiento en el proyecto. El equipo de trabajo y los miembros del proyecto al respetar las pautas establecidas obtendrán una satisfacción de logro y un proyecto completamente funcional.

La metodología tendrá como alcance el desarrollo de una versión prototipo de un botón de emergencia respetando el tiempo establecido y el cumplimiento de los objetivos planteados, logrando de esta manera cumplir con las expectativas deseadas. Para cumplir con esta finalidad se establecerán los siguientes puntos:

#### Roles

Tabla 13 Roles en el proyecto

Rol	Detalle
<b>Product Owner</b>	Julio Ricardo Proaño Orellana
<b>Scrum Master</b>	Julio Ricardo Proaño Orellana
<b>Equipo Scrum</b>	Gabriela Guamán Freddy Orquera

Nota: Tabla que contiene las responsabilidades de cada miembro que integra el proyecto.

## Reuniones

- Las reuniones generales se realizarán cada 15 días en la oficina del ingeniero responsable de manera presencial y en horarios vespertinos.
- Las reuniones con el grupo de trabajo se las realizara en horas de la tarde, por acuerdo mutuo.
- Al no contar con un establecimiento fijo de reunión para los miembros del equipo no se puede observar el avance diario con la tabla de Kanban físicamente , sin embargo, para poder establecer una comunicación diaria se usará la herramienta llamada Trello, debido a la variedad de ventajas que nos ofrece entre ellas el tablero, donde se vinculará a los integrantes y se les asignará las tareas permitiendo de este modo visualizar en cualquier momento el avance ya sea vía web o desde un dispositivo inteligente.

## Product Backlog

Tabla 14 Pila de Product Backlog

Pila de Producto					
ID	Nombre	Imp.	Pts. Estimados	Descripción	Observación
1	Diseño de Base de Datos	Alta	5	Modelar y definir las gestiones que se desean hacer sobre ellos.	
2	Módulo Logueo	Alta	4	Realizar una pantalla de ingreso a la aplicación con los campos usuario y contraseña. Adicional a esto la opción a nuevo registro.	El ingreso de datos debe ser una sola vez, posteriormente será automático.  Ingreso con datos validados

<b>3</b>	Módulo Perfil	Alta	3	Realizar una interfaz donde se podrá visualizar los datos ingresados en el registro. Si existiese cambios se los puede efectuar sin inconveniente.	Campos validados
<b>4</b>	Módulo Contactos	Alta	4	Registrar la información requerida de los contactos.	
<b>5</b>	Módulo Mensajería	Alta	5	Ingresar mensajes personalizados para su posterior envío.	
<b>6</b>	Módulo Geolocalización	Alta	5	Capturar y enviar la ubicación en tiempo real.	Verificar localización real
<b>7</b>	Módulo Alerta	Alta	5	Envío de alertas según el nivel de prioridad que desee el usuario.	Verificar cada nivel
<b>8</b>	Pruebas de Integración	Alta	5	Prueba funcional de todos los módulos.	Verificación de envío y recepción de alertas.  Comprobación de todas las funcionalidades

Nota: Tabla que contiene las historias de los módulos a realizar.

Tabla 15 Planificación de Product Backlog

	SEMANAS													
Actividades	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Módulo Logueo														
Módulo Perfil														
Módulo Mensajería														
Módulo Contactos														
Módulo Geolocalización														
Módulo Alerta														
Pruebas Finales														

Nota: Tabla que contiene la proyección de duración de cada Actividad.

Fuente 1 Gabriela Guamán y Freddy Orquera

## Planificación de Sprint



Figura 12 Estado visible de actividades para todos los usuarios

Elaborado por: Gabriela Guamán y Freddy Orquera

El equipo de Scrum hará uso del software online “TRELLO”, en el cual se identificará por columnas el estado que tiene cada Sprint. El trabajo simultaneo para cada miembro del equipo será un mínimo de 2 por lo que se dará a conocer al equipo con la abreviatura WIP=2.

### **3.2. Descripción general de requerimientos**

En este punto se describe de forma general los elementos que se consideran importantes dentro de la aplicación entre ellos se encuentran: características, restricciones, perfiles de usuarios, superposiciones y dependencias.

#### **3.2.1 Características del Prototipo**

Independientemente de la finalidad que se crea una aplicación, es importante establecer las características que la distinguen de otras y también identificar cada perfil definido en los requerimientos.

Las características son definidas con el propósito de brindar información para influir de cierto modo en la decisión del usuario al momento de optar por una opción parecida que se encuentre en la tienda de aplicaciones.

Algunas de las características que posee este prototipo son

- Uso fácil e intuitivo para una mejor comunicación bidireccional.
- El usuario obtendrá las últimas versiones de manera gratuita y actualizadas.
- Es multiplataforma y compatible con versiones antiguas.
- Utiliza pocos recursos del dispositivo, permitiendo de esta manera no perjudicar el rendimiento.
- Seguridad en la información personal del usuario.

### 3.2.2. Perfiles de usuarios

Los perfiles son identificados para establecer el emisor y receptor del mensaje de auxilio que emite la aplicación, para lo cual se establecen dos:

Tabla 16 Perfil Colaborador.

Característica	Detalle
<b>Tipo de usuario</b>	Colaborador
<b>Formación</b>	No técnica
<b>Actividad:</b>	Enviar el llamado de emergencia en una posición diferente a la del receptor.

Nota: Tabla que contiene los detalles del perfil colaborador.

Tabla 17 Perfil Suscriptor

Característica	Detalle
<b>Tipo de usuario</b>	Suscriptor
<b>Formación</b>	No técnica
<b>Actividad:</b>	Recibir el llamado de emergencia donde se visualiza el mensaje y la ubicación del solicitante.

Nota: Tabla que contiene los detalles del perfil Suscriptor.

### 3.2.3. Restricciones

La aplicación posee ciertas restricciones que se enlistan a continuación:

- El servidor no permite el almacenamiento mayor a 1G por persona.
- El prototipo funciona solo en dispositivos inteligentes que dispongan GPS.
- El usuario debe mantener conexión a internet para poder enviar y recibir las alertas de emergencia.

- Los usuarios no podrán poseer más de una cuenta con el mismo correo electrónico.
- El usuario debe dar permiso de acceso al teléfono y a su ubicación, permitiendo de esta manera obtener las coordenadas de latitud y longitud que se envían en el llamado.
- La actualización debe ser realizada con conexión a internet y tener un espacio suficiente.

#### **3.2.4. Superposiciones y dependencias**

El servidor que se usara es Firebase, el cual brindara las siguientes funciones:

- Base de datos NoSQL que se permite una sincronización de datos en tiempo real, ya que se encuentra alojada en la nube.
- Almacenamiento con capacidad de alojar datos para perfil de usuario, contactos, mensajes y ubicación.
- Conexión estable incluso en momento de desconexión del cliente los datos serán actualizados sin volver a ingresarlos.
- Se asume que el prototipo tiene adaptabilidad para todo tipo de dispositivo y plataforma móvil.
- Se asume que la velocidad y rendimiento de la aplicación depende de las características del dispositivo.

Los puntos descritos son un factor fundamental, sin embargo, como punto adicional y relevante en el prototipo son los requisitos de rendimiento que brinde la aplicación al usuario.

### **3.3. Requisitos de rendimiento**

La estructura de la aplicación en los equipos debe tener el mayor rendimiento en los equipos por lo que se define algunos requisitos:

- No consumir más del 50% del rendimiento del dispositivo móvil en el momento de realizar los llamados de emergencia.
- Las transacciones que se realicen deben tener un tiempo mínimo de repuesta de 1 minuto.
- El 99% de los procesos se realizarán con éxito, teniendo en cuenta que el 1% serán las alertas enviadas sin conexión de red.
- Garantizar la interacción simultanea de varios usuarios en la aplicación.

Adicionalmente, la mayoría de aplicaciones móviles contienen dos puntos relevantes que hacen la diferencia y llaman la atención al usuario. Estas son:

#### **Seguridad**

La seguridad del prototipo es establecida de la siguiente manera:

- Usar contraseñas para cada usuario permitiendo tener acceso a la aplicación solo las personas que poseen una autorización.
- Manipulación de datos de manera segura sin afectar el rendimiento o perdida de información en el proceso.
- Encriptación de contraseñas para no poder ser adulterada o suplantar la identidad por otro usuario.
- Ingreso y uso del sistema sin intervención de aspectos adicionales como uso de carpetas o archivos personales del dispositivo.
- Ubicación no compartida con otros usuarios al no ser enviado por el usuario.



## **Portabilidad**

Al manejar herramientas libres y el desarrollo en multiplataforma garantiza una portabilidad exitosa, ya que en está interviniente la simplicidad de instalación en cualquier dispositivo y la manipulación de todas las funciones sin importar la plataforma.

En cuanto al tamaño y pixeles de pantalla, al ser una aplicación adaptativa no afecta el diagonal de pantalla en el que se ejecute la aplicación.

### **3.4. Interfaces**

La interfaz es una capa visible donde permite la comunicación entre el usuario y la aplicación, cada interfaz tiene una composición diferente con elementos como: imágenes, iconos, botones, texto y otros. Existen diferentes interfaces con requerimientos específicos que permiten un funcionamiento adecuado, las más significativas son la interfaz de usuario y de comunicación.

#### **3.4.2. Usuario**

La interfaz de usuario se la define como la interacción con un dispositivo móvil, generalmente para ser aceptada por el usuario final debe ser fácil e intuitiva para evitar una inversión de tiempo en aprender a manejar los pasos necesarios para obtener el resultado deseado. Otro aspecto importante es la cantidad de pantallas que intervienen, por lo cual es ideal solo mostrar al usuario lo que necesita ver simplificando pantallas innecesarias que puedan crear confusión.

Una interfaz interactiva lo vuelve más atractivo, el comportamiento del aplicativo debe permitir familiarizarse de tal manera que se entienda la finalidad y la variedad de características que nos puede ofrecer. En la figura 13 se puede observar el proceso que se llevó a cabo para la elaboración del diseño de interfaz.

### Diagrama de elaboración de interfaz

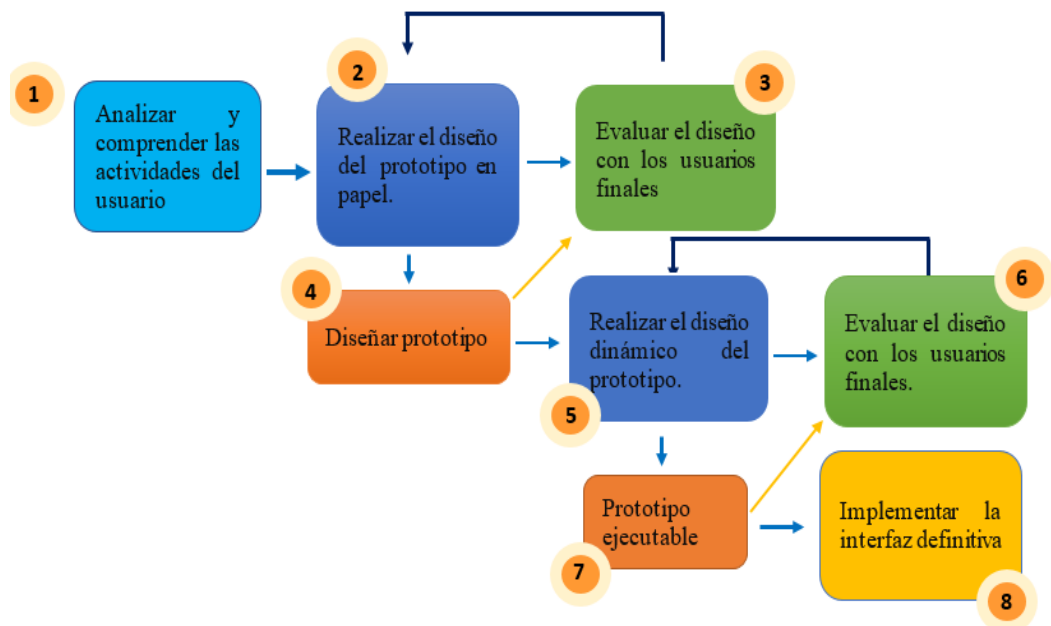


Figura 13 Representación de la guía para llegar a la interfaz final.  
Elaborado por: Gabriela Guamán y Freddy Orquera

A continuación, se detallan los puntos que intervienen en el diagrama de elaboración de la interfaz.

- **Paso 1 - Analizar y Comprender las actividades del usuario:** En el primer punto se establece la finalidad de la aplicación.
- **Paso 2 -Realizar el diseño del prototipo en papel:** Se realiza un boceto utilizando materiales básicos donde se capturar de una forma rápida y flexible la funcionalidad del sistema.
- **Paso 3 -Evaluar el diseño con los usuarios finales:** Este punto evalúa el diseño que dará solución al problema planteado por el usuario.
- **Paso 4- Diseñar prototipo:** Se trabaja en un software de diseño donde se plasmará el diseño aprobado tomará color, dependiendo del usuario se puede tener un patrón de colores que representen a la empresa destinada. En este

punto el diseño será nuevamente presentado al interesado y según el resultado se podrá volver a diseñar las interfaces no aprobadas o modificadas o a su vez pasar al siguiente punto.

- **Paso 5 - Realizar el diseño dinámico del prototipo:** El prototipo se realiza de forma interactiva donde el usuario podrá navegar simulando la aplicación.
- **Paso 6 -Evaluar el diseño con los usuarios finales:** En este punto se evalúa nuevamente el diseño con las modificaciones que se han ido realizando.
- **Paso 7 - Prototipo ejecutable:** El sistema será ejecutado en un emulador donde se podrá observar la adaptación del prototipo en diferentes tamaños de pantallas y plataformas.
- **Paso 8 - Implementar la interfaz definitiva:** Al ser aprobado por el usuario, comenzara el proceso de desarrollo de funcionalidades en cada interfaz.

### 3.4.3. Comunicación

El prototipo utiliza una comunicación en tiempo real por lo que los datos son almacenados en la nube en tipo JSON, donde es posible incrementar reglas que permitan solicitudes HTTP a través de token o simplemente desde una URL.

La figura 14 describe la comunicación que realiza al momento de ejecutar por primera vez o realizar una actualización en la base de datos para continuamente ser almacenada en la nube y posteriormente notificar en milisegundo a todos los dispositivos de manera simultánea. Otro de los beneficios es la optimización que ofrece el SDK de firebase, pues al perder la conexión almacena la información en la cache del dispositivo permitiendo una sincronización al momento de recuperar la conexión a la red.

Conexión del servidor con la aplicación.



Figura 14 Representación de la conexión de un servidor a todos los usuarios.  
Elaborado por: Gabriela Guamán y Freddy Orquera

### 3.5. Requerimientos

En el desarrollo del prototipo se encuentra el punto de categorización, donde se divide los requerimientos en dos conceptos diferentes funcionales y no funcionales.

#### 3.5.2. Requerimientos funcionales

En este punto se describe las actividades que el sistema debe proporcionar generalmente se encuentra asociado a las entradas o salidas de procesos.

Las siguientes tablas muestran detalladamente los requerimientos solicitados por el usuario para el uso de la aplicación.

Tabla 18 Registro de usuario

Requerimiento Funcional N°: 1	
<b>Usuario:</b> Usuario General	
<b>Nombre de requerimiento:</b> Registro de usuario	
<b>Prioridad del requerimiento:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Requerimiento NO funcional:</b> RNF 1 RNF 2 RNF 4	
<b>Descripción:</b> Almacenar y crear un id de usuario nuevo.	
<b>Observaciones:</b> Los usuarios que deseen hacer uso de la aplicación deberán registrar la información solicitada con un perfil establecido para posteriormente iniciar sesión.  Si la aplicación es abierta por primera vez se debe ingresar las credenciales solicitadas en la autenticación de usuario caso contrario la aplicación iniciara tomando los datos almacenados.	

Nota: Esta tabla detallada el requerimiento de registro de usuarios para el uso de la aplicación.

Tabla 19 Consultar datos de usuario

Requerimiento Funcional N°: 2	
<b>Número:</b> 2	<b>Usuario:</b> Usuario General
<b>Nombre de requisito:</b> Consultar datos de usuario	
<b>Prioridad del negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Requerimiento NO funcional:</b> RNF 1 RNF 2 RNF 3 RNF 4	

<b>Descripción:</b>
Extraer y mostrar información de la base de datos.
<b>Observaciones:</b>
Los usuarios ya registrados podrán observar los datos ingresados e inclusive editarlos si se hubiese cometido algún error o actualizar la información las veces que desee el usuario.

Nota: Esta tabla detallada el requerimiento de consulta de datos de usuario, donde se podrá ingresar en cualquier momento.

Tabla 20 Registro de contactos

Requerimiento Funcional N°: 3	
<b>Usuario:</b> Usuario General	
<b>Nombre de requisito:</b> Registro de contactos	
<b>Prioridad del negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Requerimiento NO funcional:</b> RNF 1 RNF 2 RNF 3 RNF 4	
<b>Descripción:</b> Almacenar información en la base de datos sección contactos.	
<b>Observaciones:</b> El usuario podrá realizar el ingreso de contactos que crea conveniente no existe un número límite, los datos serán utilizados para enviar la alerta en un llamado de emergencia.	

Nota: Esta tabla detallada el requerimiento de registro de contactos para el envío de alertas.

Tabla 21 Consulta de Contactos de Emergencia

Requerimiento Funcional N°: 4
<b>Usuario:</b> Usuario General
<b>Nombre de historia:</b> Consulta de Contactos de Emergencia

<b>Prioridad del negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Requerimiento NO funcional:</b> RNF 1 RNF 2 RNF 3 RNF 4	
<b>Descripción:</b> Extraer información de la base de datos.	
<b>Observaciones:</b> El usuario tendrá acceso a la pantalla de contactos, donde visualizará los datos ingresados, los mismos que podrán ser actualizados si los desea o simplemente observar la información ingresada.	

Nota: Esta tabla detallada el requerimiento de consulta de los datos ingresados en los contactos de emergencia en caso de existir alguna actualización o eliminación del mismo.

Tabla 22 Mensajería personalizada o predefinida

Requerimiento Funcional N°: 5	
<b>Usuario:</b> Usuario General	
<b>Nombre de historia:</b> Mensajería personalizada o predefinida	
<b>Prioridad del negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Requerimiento NO funcional:</b> RNF 1 RNF 2 RNF 4	
<b>Descripción:</b> Almacenar mensajes para un posterior uso.	
<b>Observaciones:</b> El usuario al momento de enviar el llamado de emergencia dependiendo del nivel de alerta que seleccione, se mostrará una pantalla donde podrá seleccionara un mensaje sea personalizado o predefinidos que identifique la situación en la que se encuentre.	

Nota: Esta tabla detallada el requerimiento de mensajería, uno de los más significativos ya que serán utilizados para el envío de alertas que realice el usuario.

Tabla 23 Geo localización

Requerimiento Funcional N°: 6	
<b>Usuario:</b> Usuario General	
<b>Nombre de historia:</b> Geo localización	
<b>Prioridad del negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Requerimiento NO funcional:</b> RNF 1 RNF 2 RNF 3 RNF 4	
<b>Descripción:</b> Obtener permisos del dispositivo y extraer puntos en tiempo real.	
<b>Observaciones:</b> El usuario al momento de enviar el llamado de emergencia independientemente del nivel de alerta que seleccione el usuario el mensaje captura las coordenadas de ubicación en tiempo enviado de este modo al receptor y así poder visualizar en el mapa del contacto junto con él mensaje enviado.	

Nota: Esta tabla detallada el requerimiento de Geo localización que se lo realiza en tiempo real.

### 3.5.3. Requerimientos no funcionales

En este punto se especifica las propiedades que contiene la aplicación, estas se producen para satisfacer las necesidades del usuario.

Tabla 24 Flexibilidad

Requerimiento NO Funcional N°: 1
<b>Usuario:</b> Usuario General
<b>Nombre de historia:</b> Flexibilidad
<b>Prioridad:</b> Alta



<b>Descripción:</b>
Realizar entornos y componentes reusables.
<b>Observaciones:</b>
La aplicación acepta cambios a medida del uso que le brinda el cliente.

Nota: Esta tabla detallada el requerimiento NO funcional de Flexibilidad.

Tabla 25 Seguridad

Requerimiento NO Funcional N°: 2
<b>Usuario:</b> Usuario General
<b>Nombre de historia:</b> Seguridad
<b>Prioridad:</b> Alta
<b>Descripción:</b>
Datos recibidos y extraídos de la base de datos serán encriptados.
<b>Observaciones:</b>
La información del usuario es encriptada previniendo de esta manera la difusión de datos de usuario en lugares externos para usos no aceptados.

Nota: Esta tabla detallada el requerimiento NO funcional de Seguridad.

Tabla 26 Mantenibilidad

Requerimiento NO Funcional N°: 3
<b>Usuario:</b> Usuario General
<b>Nombre de historia:</b> Mantenibilidad
<b>Prioridad:</b> Alta
<b>Descripción:</b>
Documentar código y cambios realizados en la aplicación.
<b>Observaciones:</b>
La habilidad de manejar la aplicación para modificar errores o atributos solicitados sin perjudicar el rendimiento.

Nota: Esta tabla detallada el requerimiento NO funcional de Mantenibilidad.

Tabla 27 Usabilidad

Requerimiento NO Funcional N°: 4
<b>Usuario:</b> Usuario General
<b>Nombre de historia:</b> Usabilidad
<b>Prioridad:</b> Alta
<b>Descripción:</b>  Realizar una interfaz intuitiva para el usuario.
<b>Observaciones:</b>  La interfaz ha sido diseñada para que el usuario pueda interactuar fácilmente, los botones y campos de texto son lo suficientemente grandes.

Nota: Esta tabla detallada el requerimiento NO funcional de Usabilidad.

### 3.6. Diseño

A continuación, se detalla el funcionamiento de los diagramas UML (Unified Modeling Language), donde se describe la comunicación, comportamiento, interacción, estructura y diseño que interviene en el diseño y desarrollo de la aplicación.

#### 3.6.1. Diagrama de caso de uso

La figura 15 muestra una interpretación de las actividades que puede realizar un actor en la aplicación.

## Caso de uso de la Aplicación

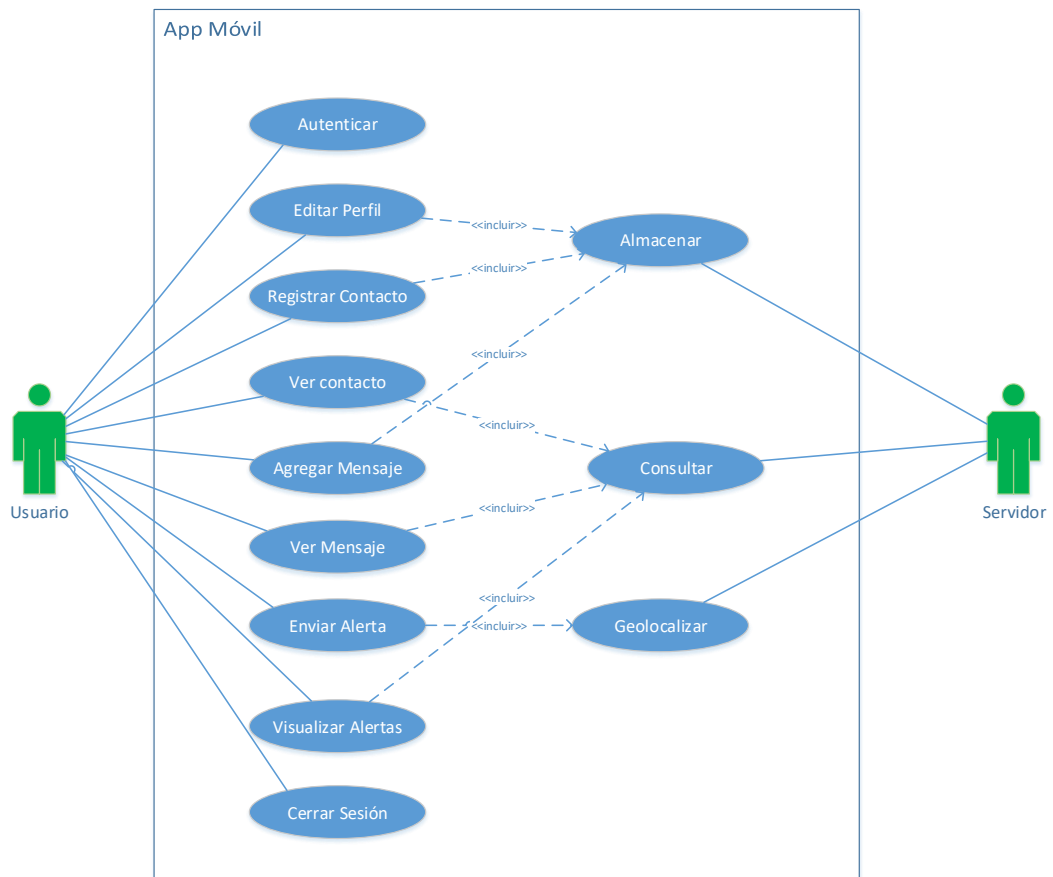


Figura 15 Diagrama de caso de uso de la aplicación móvil  
Elaborado por: Gabriela Guamán y Freddy Orquera

Tabla 28 Ficha de caso de uso de la aplicación móvil

Nombre	Aplicación Móvil
<b>Autor:</b>	Gabriela Guamán y Freddy Orquera
<b>Fecha:</b>	22/05/2020
<b>Descripción:</b>	Permite al actor usuario: autenticar, edición de perfil, registro de contacto, creación de mensajes personalizados, creación de alerta y cierre de sesión  El actor servidor: almacena, sincroniza y geo localiza.
<b>Actores:</b>	Usuario

<b>Precondiciones:</b>	El usuario debe realizar un correcto logeo para ser uso total de la aplicación.
<b>Flujo Normal:</b>	<ul style="list-style-type: none"> <li>• El primer ingreso del usuario se verifica con la autenticación.</li> <li>• El usuario puede editar su perfil, registrar nuevos contactos, agregar mensajes personalizados, enviar alertas.</li> <li>• La aplicación realiza todos los ingresos y consultas a la base de datos.</li> <li>• La geolocalización obtiene los puntos del usuario en tiempo real.</li> </ul>
<b>Flujo Alternativo:</b>	<ul style="list-style-type: none"> <li>• El ingreso de datos erróneos es comunicado al usuario para su corrección.</li> <li>• En la instalación se solicita al usuario el permiso de acceso al GPS del dispositivo.</li> </ul>
<b>Postcondiciones:</b>	Las consultas son visualizadas en pantalla.

Nota: La tabla describe el diagrama de caso de uso de la aplicación.

### 3.6.2. Diagrama de secuencia

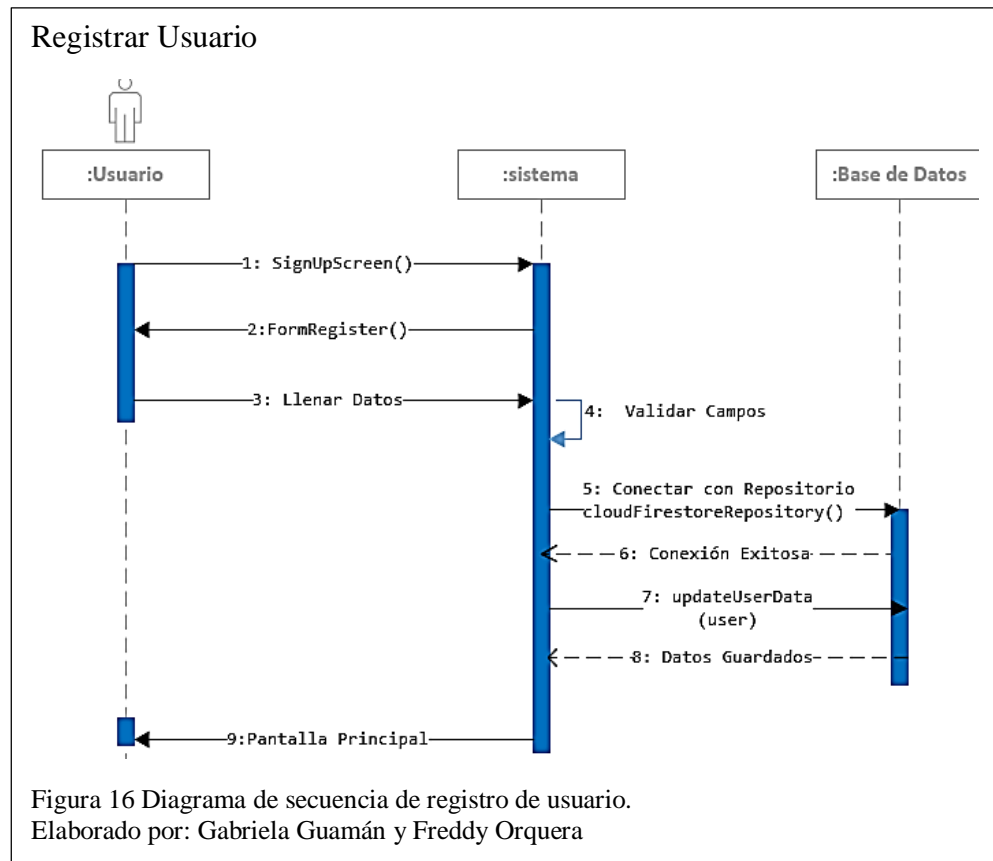
Se definen como la interacción dinámica entre los objetos que interactúan en la aplicación. A continuación, se representan los diagramas para los diferentes casos.

#### • Diagrama de secuencia –Registrar Usuario

Como muestra el diagrama de la figura 16, el proceso tiene la siguiente esquematización:

1. El usuario, sin necesidad de poseer conocimientos en tecnología inicia la aplicación donde al cargarse muestra una pantalla con la siguiente estructura: en la parte superior cuenta con dos campos de autenticación mientras que la parte inferior tiene dos botones uno de ingreso y otro de registro, el usuario presionara el apartado de registro activando de este modo el método SignUpScreen().

2. Una vez recibido el llamado del método, la aplicación muestra un formulario de registro con los siguientes campos: nombre, correo, contraseña, dirección, teléfono, fotografía (opcional).
3. El usuario llena todos los campos solicitado por el sistema y agrega una foto de identificación a su perfil esta puede ser añadida en el momento de registro o posteriormente, una vez completa la información se presionará el botón “Registarse”.
4. Al recibir la acción y con el formulario lleno el sistema procede a validar los campos ingresados considerando que no todos tienen el mismo tipo de dato, por ende, en un campo numérico no ingresan letras y en un campo alfabético solo se permite letras ya sean mayúsculas o minúsculas.
5. Una vez finalizada la validación de campos la aplicación realiza un enlace para emitir una conexión desde el dispositivo al repositorio de la base de datos.
6. La base de datos responde con un mensaje “conexión exitosa”, dando a interpretar que el servicio se encuentra disponible para interactuar con la aplicación.
7. Creada la conexión, la aplicación procederá a ejecutar el método `updateUserData`, la cual permite tomar la información enviada desde el dispositivo y almacenarla en la base de datos.
8. Recibida la información la base de datos emite una respuesta representativa con el mensaje de “datos guardados”, esta dará a conocer que la información fue almacenada de forma correcta.
9. Una vez finalizado y comprobado el almacenamiento la aplicación muestra la página principal.

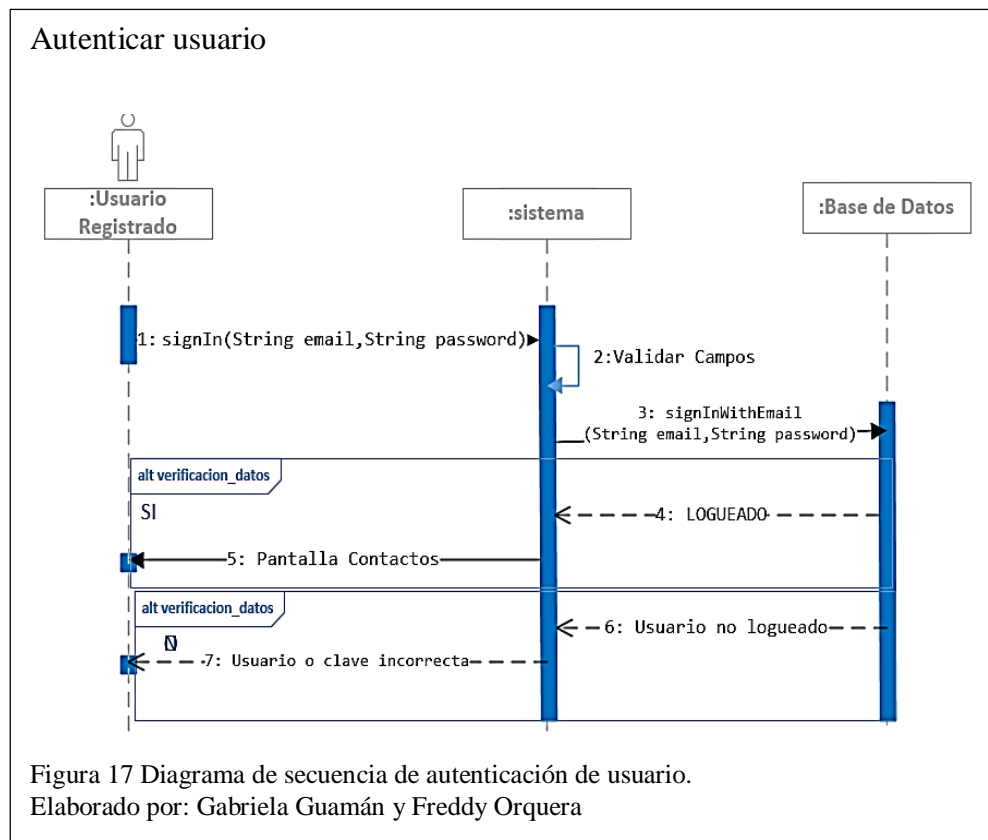


- **Diagrama de secuencia –autenticar**

Una vez registrado la aplicación muestra la pantalla de Login como muestra en la figura 17, donde realiza las siguientes acciones:

1. El usuario al ya poseer un registro en la base de datos procede a ingresar los datos solicitados que son: email y contraseña.
2. Al tener los campos llenos la aplicación procede a validar los campos para poder ser enviados.
3. Una vez finalizada la validación, la aplicación ejecuta el método “singInWithEmail” que conecta con la base de datos para verificar la información ingresada en el registro, donde la respuesta de dicha consulta plantea dos alternativas.

4. La primera opción es una respuesta afirmativa que se identifica con el mensaje “Logueado”, en la cual confirma que el usuario se encuentra registrado y los datos recibidos son correctos.
5. Una vez recibida la respuesta de confirmación la aplicación muestra al usuario la pantalla de contactos y la barra de menú para uso libre y total de la aplicación.
6. Como segunda opción es una respuesta negativa que es identificada con el mensaje “Usuario no logueado”, pues los datos enviados fueron erróneos o el usuario no se encuentra registrados en la base de datos.
7. Al no ser admitido las credenciales el sistema presenta un aviso personalizado con el mensaje “Usuario o clave incorrecta” indicando de este modo al usuario el posible error.



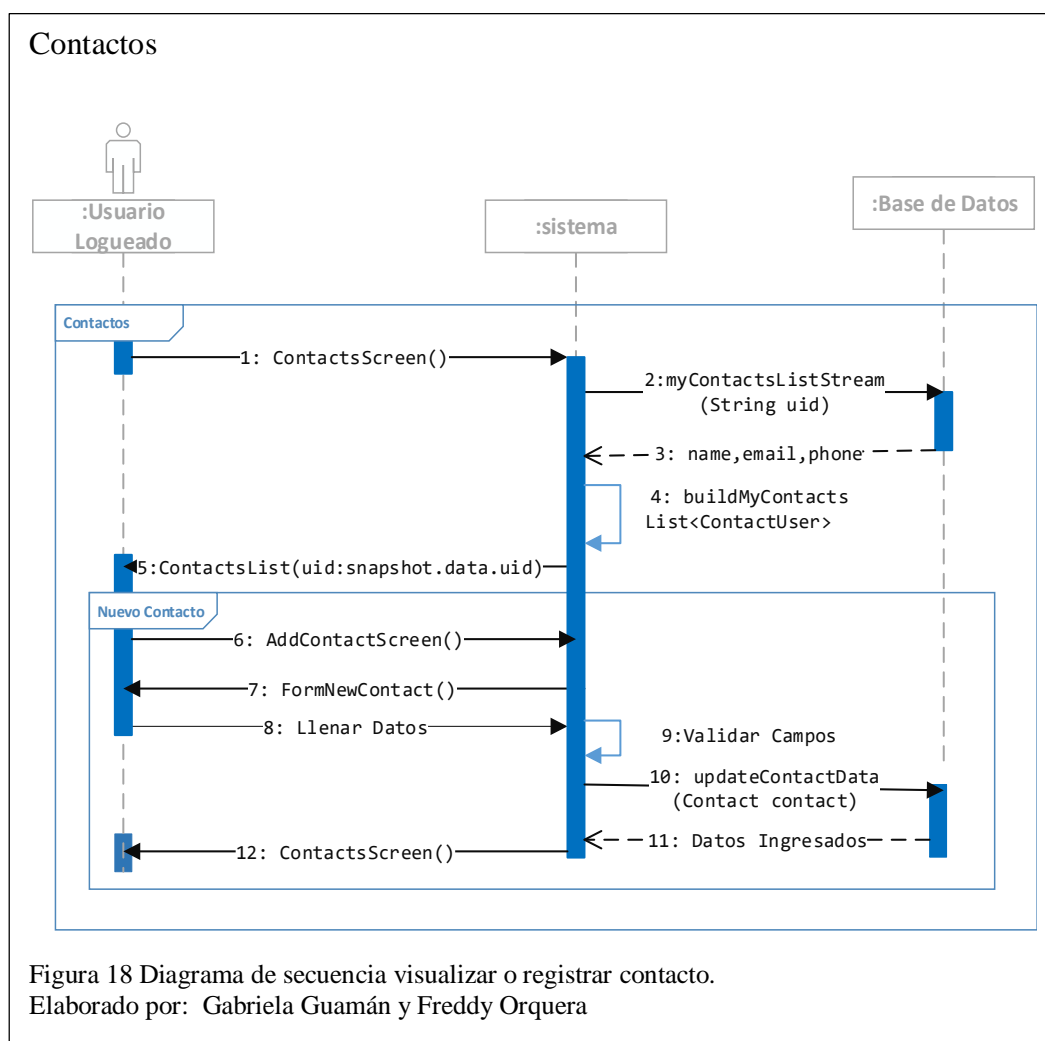
- **Diagrama de secuencia – Contactos**

La figura 18 muestra el diagrama de secuencia de la interfaz de contactos. A continuación, se detallan los pasos a seguir:

1. El usuario al iniciar sesión de manera satisfactoria automáticamente el usuario convoca la clase `ContactsScreen`, donde se visualizará la interfaz de contactos.
2. La aplicación solicita información a la base de datos mediante la clase `myContactsListStream`, la cual envía el id del usuario para consultar los registros almacenados.
3. La base de datos retorna el nombre, email y teléfono de contacto registrados del id solicitado en la consulta.
4. Una vez recibido los datos la aplicación ejecuta internamente el método `buildMyContacts`, el cual realiza la construcción de una lista de contactos.
5. Finalizado el proceso anterior la aplicación muestra en pantalla la lista de contactos registrados de manera ordenada y legible para el usuario, en caso de ser nuevo la lista se encontrará vacía.
6. En la parte inferior se encuentra la opción de “Nuevo Contacto” donde se puede registrar ilimitado número de contactos. En este caso al presionarlo el usuario activa una subpantalla.
7. Una vez activada la pantalla de registro la aplicación mostrara un formulario solicitando información necesaria para el envío de alerta.
8. El usuario debe llenar obligatoriamente toda la información que solicita cada campo.
9. La aplicación valida todos los campos verificando la concordancia entre el campo y el tipo de dato ingresado.



10. Finalizado el proceso de validación la información es enviada a la base de datos para ser almacenada e inmediatamente colocada en el registro de contactos.
11. Una vez terminado el proceso de almacenamiento la base de datos envía a la aplicación un mensaje de confirmación de proceso.
12. Para finalizar la aplicación regresa a la pantalla principal de contactos.

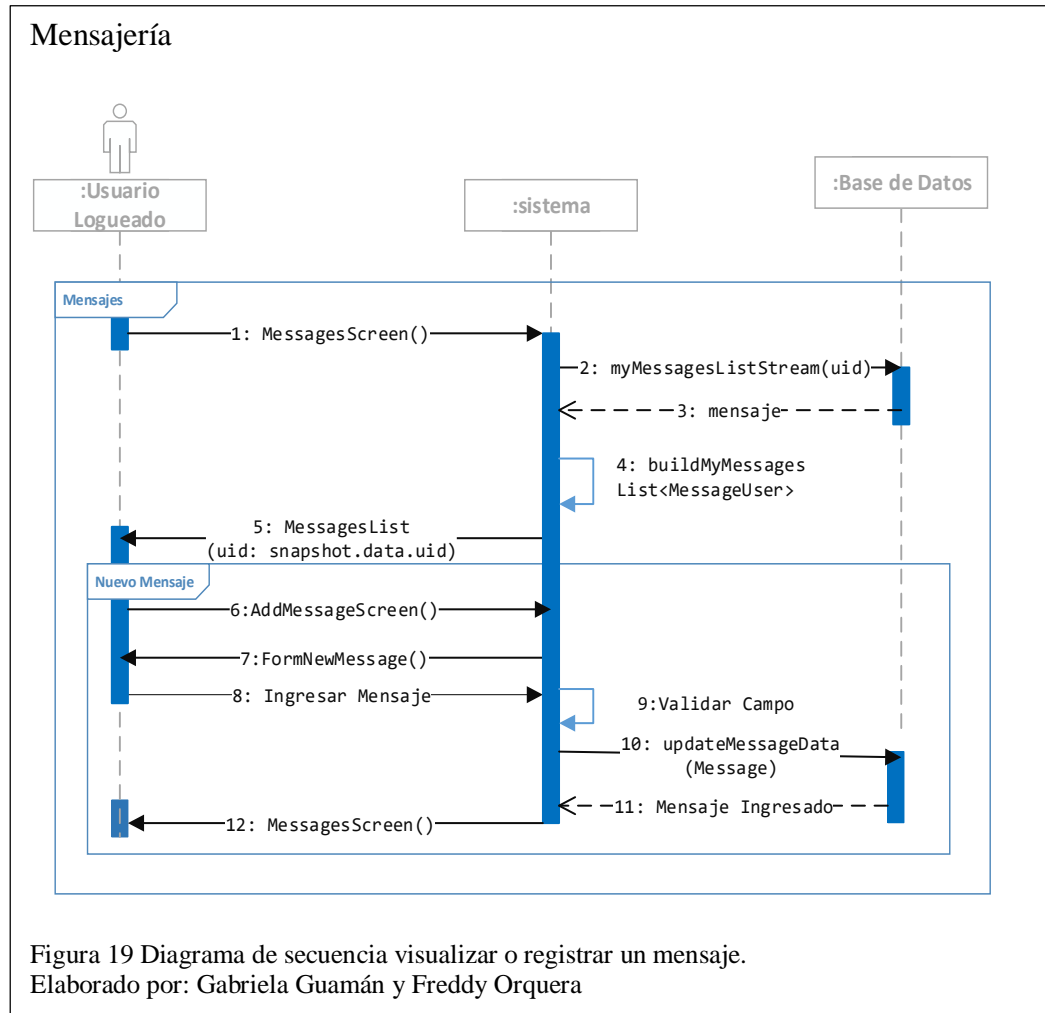


### • Diagrama de secuencia –Mensajería

La figura 19 representa la interacción entre el usuario y la aplicación, pues el usuario registrado podrá visualizar y crear nuevos mensajes. A continuación, se detalla los pasos:

1. El usuario presiona el icono de mensajería donde activa el llamado a la pantalla de mensajería.
2. La aplicación solicita información a la base de datos mediante la clase `myMessagesListStream`, esta envía el id del usuario para consultar el o los mensajes registrados.
3. La base de datos devuelve los mensajes registrados según el id de usuario solicitado en la consulta.
4. Una vez finalizada la carga de mensajes la aplicación automáticamente procede a listar ordenadamente la información.
5. Finalizado el proceso anterior, la aplicación muestra en pantalla en la parte superior los mensajes registrados de manera rápida.
6. El usuario presiona el botón de “Nuevo” donde activa inmediatamente un llamado de nuevo registro al sistema.
7. La aplicación recibe el llamado y presenta un formulario que posee un solo campo donde se ingresará el mensaje personalizado.
8. El usuario debe ingresar un mensaje este puede incluir números y signos de exclamación.
9. La aplicación realiza la validación internamente del campo de mensaje, pues no podrá estar vacío.
10. Cuando el proceso ha sido finalizado el mensaje es enviado a la base de datos, donde es almacenado según el usuario que corresponde.
11. La base de datos envía un mensaje de información indicando el estado del proceso realizado “Mensaje ingresado” este no será visible para los usuarios.

12. Finalizado el almacenamiento y confirmado el ingreso a la base de datos la aplicación retorna al usuario automáticamente a la pantalla principal de mensajes.



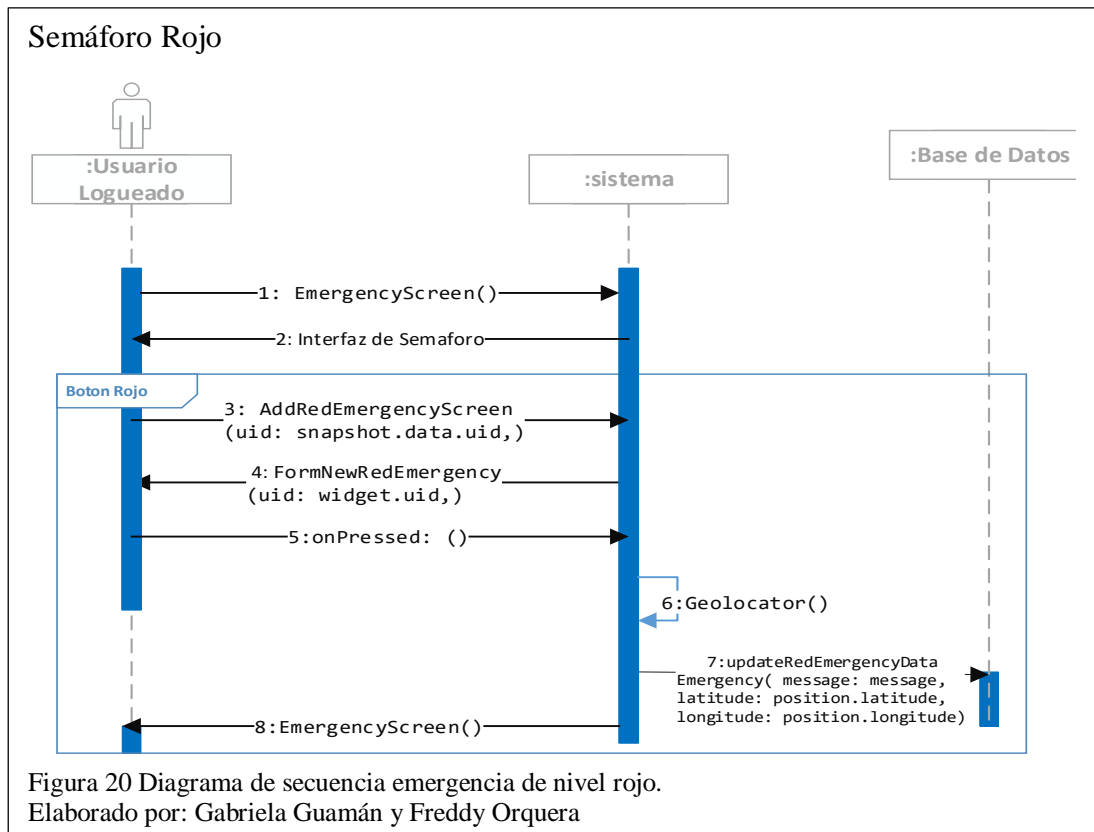
### • Diagrama de secuencia –Botón de Pánico

El usuario tendrá acceso al semáforo de pánico, el cual se encuentra establecido por 3 colores principales que representan el grado de emergencia en el que se encuentra el usuario.

#### ▪ Botón Rojo

En la figura 20 se muestra el diagrama de secuencia de la interacción entre el usuario y la aplicación los pasos que se realiza son los siguientes:

1. El usuario accede a la interfaz de emergencia mediante el icono de home que se encuentra en la parte inferior de la aplicación.
2. Esta pantalla presenta una semaforización donde cada botón representa un tipo de emergencia.
3. En este caso el usuario presiona el botón rojo que capta el id del usuario y representa un nivel de peligro alto.
4. Inmediatamente se muestra una subinterfaz con un formulario que consta de una lista de mensajes almacenados ordenados y con casilleros de selección.
5. Una vez que el usuario haya seleccionado el mensaje que identifique la situación o sentimiento que se encuentre atravesando en ese momento se presiona “Enviar”
6. La aplicación captura automáticamente la referencia de la posición en tiempo real tanto en latitud como longitud.
7. La información recopilada que consta del mensaje y los puntos tomados por la geolocalización son enviados a la base de datos para su almacenamiento en la tabla correspondiente.
8. Para finalizar el usuario es retornado a la interfaz principal que es la semaforización

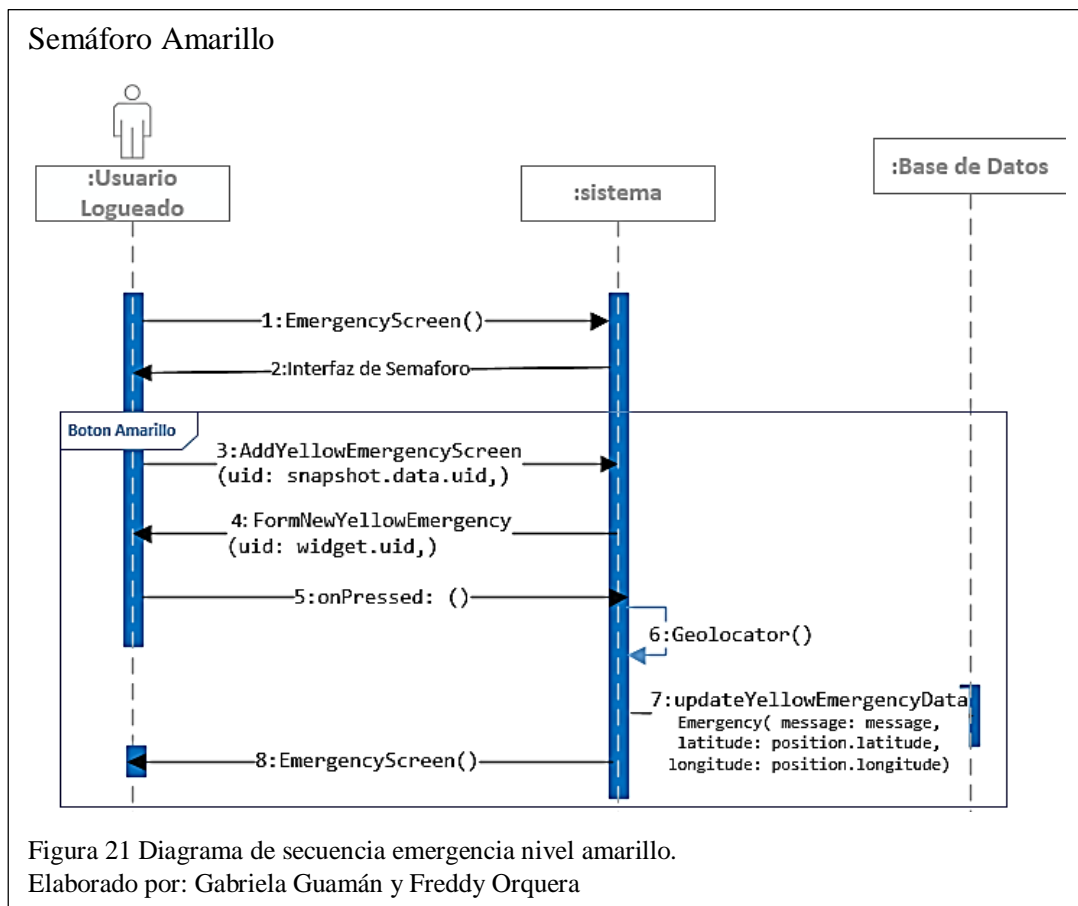


### ▪ Botón Amarillo

En la figura 21 muestra el diagrama de secuencia del nivel de alerta color amarillo, A continuación, se detalla los pasos:

1. El usuario accede a la interfaz de emergencia mediante el icono situado en la barra inferior de la aplicación esta activa un llamado a la pantalla.
2. La aplicación responde la petición y muestra una pantalla con 3 botones que representan la semaforización cada uno con un nivel de emergencia.
3. Para este caso el usuario presiona el botón amarillo que representa un tipo de emergencia de nivel intermedio.
4. La aplicación despliega un formulario que incluye una lista de mensajes y contactos registrados ordenados cada uno con un cuadro selector para mayor facilidad.

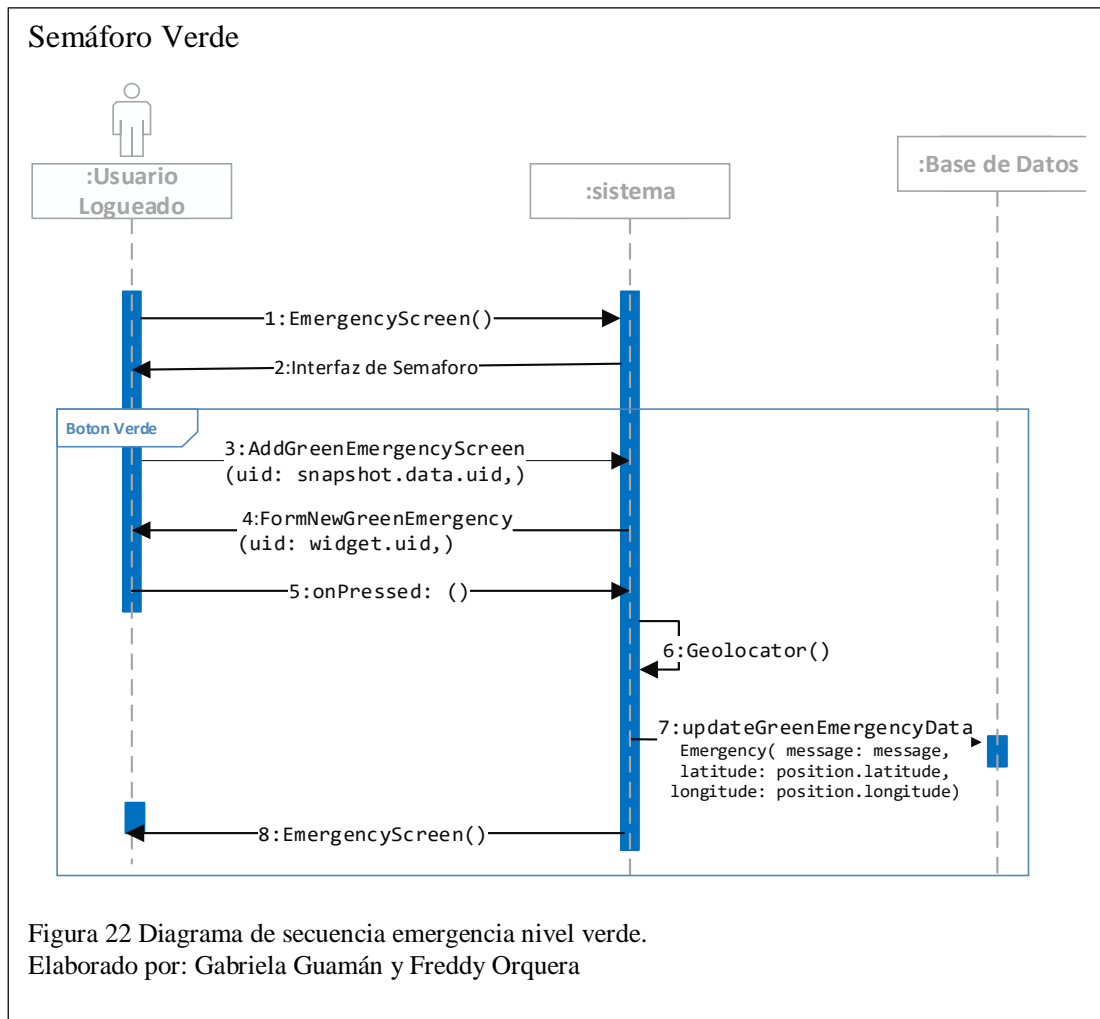
5. Una vez seleccionado la información que consta de un mensaje y el contacto al que se desea enviar la alerta el usuario presiona “Enviar.
6. La aplicación ejecuta internamente y de forma automática el geolocalizador, este capta los puntos en tiempo real.
7. Los mensajes seleccionados junto con la localización tomada por el dispositivo son enviados a la base de datos para su almacenamiento en la tabla de emergencia.
8. Para finalizar la aplicación muestra al usuario la interfaz de la semaforización.



#### ▪ Botón Verde

La figura 22 muestra el diagrama de secuencia de un envío de alerta nivel bajo y definida con el botón color verde. A continuación, se describe cada paso.

1. El usuario accede a la interfaz de emergencia mediante el icono de home situado en la parte inferior de la aplicación.
2. Seguidamente se presenta la interfaz de semaforización, cada color representa a un tipo de emergencia.
3. Para este caso se presiona el botón verde que simboliza un tipo de emergencia de nivel bajo.
4. La aplicación muestra un formulario que incluye un cuadro de texto para poder ingresar el mensaje a libre expresión, seguidamente se lista los contactos registrados junto con su casillero de selección.
5. El usuario una vez seleccionado el contacto de envió junto con el mensaje personalizado presiona el botón de “Enviar”.
6. La aplicación internamente ejecuta de forma automática la captura de datos en tiempo real de longitud y latitud del usuario.
7. La información generada y capturada son empaquetados para ser enviado a la base de datos y almacenarla en la tabla de emergencias.
8. Inmediatamente la aplicación realiza una recarga automática a la página principal de semaforización.



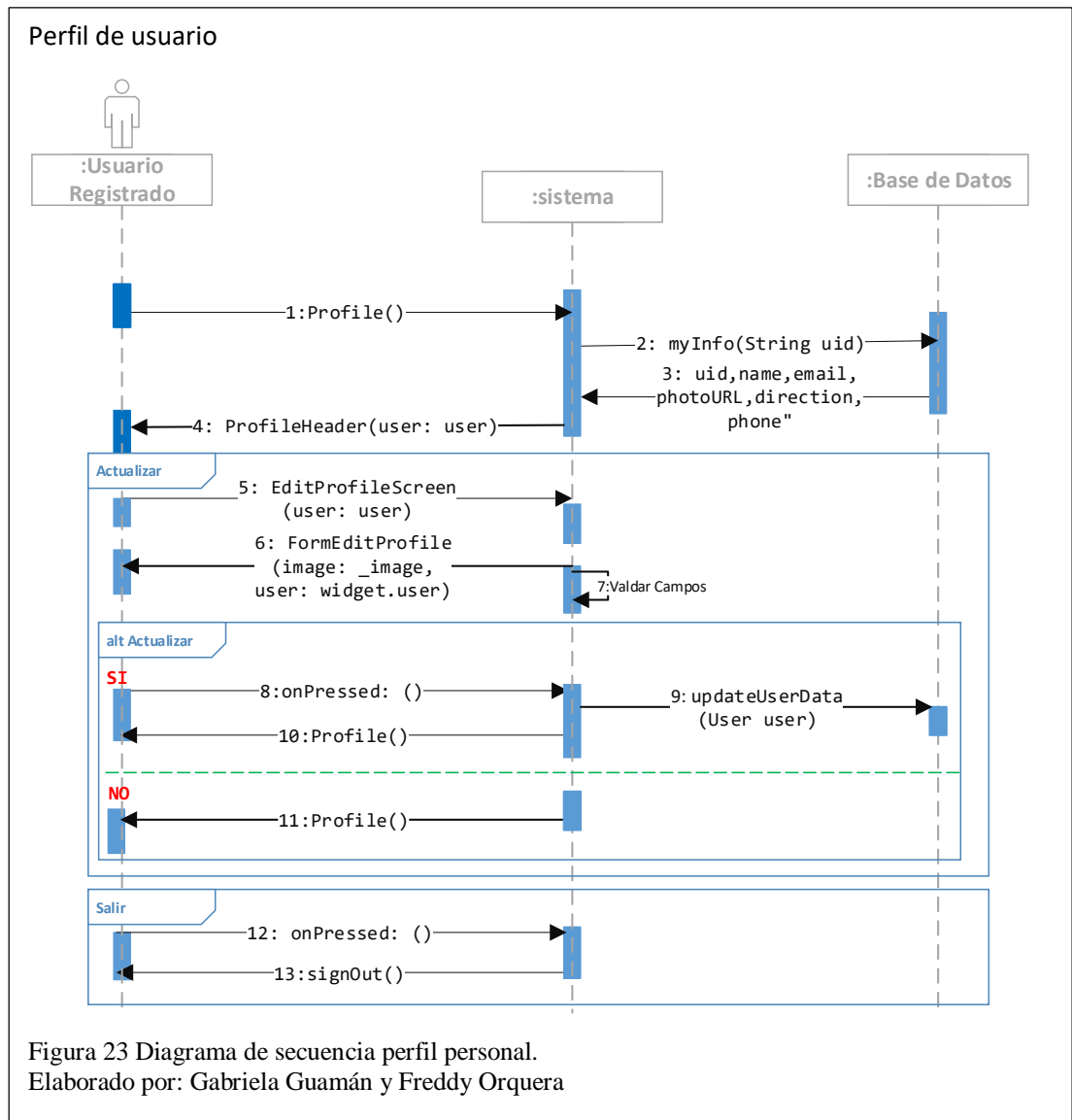
- **Diagrama de secuencia –Perfil**

La figura 23 representa un diagrama de secuencia de perfil este contiene pasos que se detallan a continuación:

1. El usuario registrado ingresa a la interfaz de perfil personal mediante el icono de ajustes que se encuentra en la barra inferior.
2. La aplicación solicita información almacenada en la base de datos del usuario mediante la id.
3. La base de datos responde a la solicitud enviando la información requerida de manera ordenada y legible.



4. Al poseer toda la información se presenta en el parte superior la fotografía del usuario, el nombre y correo electrónico ingresado en el registro.
5. El usuario puede seleccionar la alternativa de actualizar la información que el crea conveniente.
6. La aplicación presenta al usuario una subpantalla con todos los campos llenos correspondiente a la información que fue ingresada en el registro.
7. Realiza una validación de campos a todo el formulario, se debe tener en cuenta que los campos no deben estar vacíos.
8. En la parte inferior de la interfaz muestra el botón de “Editar”, esta opción es presionada por el usuario.
9. Inmediatamente la información modificada es enviada a la base de datos con el id del usuario para mejor precisión del proceso.
10. La aplicación muestra inmediatamente la pantalla de perfil de usuario principal con los cambios realizados.
11. , Al no poseer modificaciones en los datos simplemente la aplicación regresa de forma automática a la pantalla principal de perfil.
12. Finalmente, en la parte inferior como segunda opción se tiene el cierre de sesión de la aplicación donde el usuario puede presionar si lo desea.
13. Al recibir el evento anterior la aplicación muestra una automáticamente la interfaz principal de login.

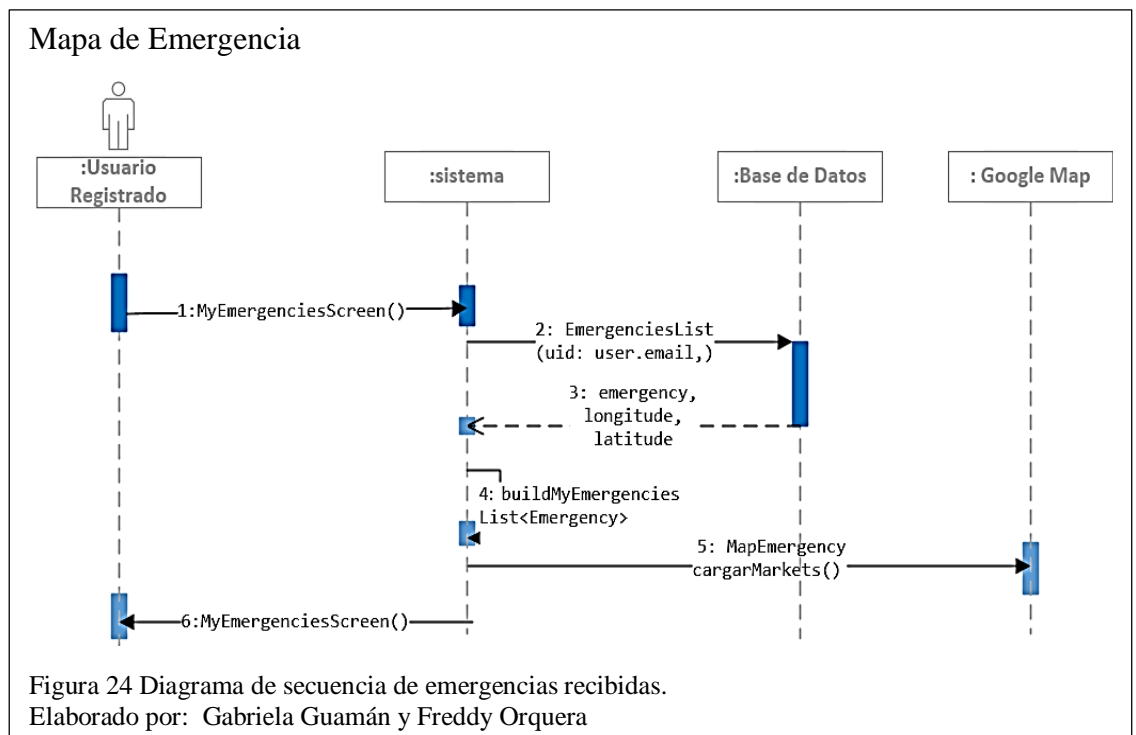


### • Diagrama de secuencia –Mapa

La figura 24 representa el diagrama de secuencia del mapa donde está interacción con la aplicación se detalla a continuación:

1. El usuario registrado ingresa a la pantalla “Mapa” esta se puede acceder por medio del icono situado en la barra inferior de la aplicación.
2. Para cargar la información el sistema envía a la base de datos una petición de información del usuario, esta información es solicitada mediante el id del correo electrónico.

3. La base de datos envía la respuesta de la información con los datos de mensaje, longitud y latitud.
4. El sistema recibe los datos y de manera automática alberga en una lista toda la información recopilada.
5. Continuamente esta lista es enviada al mapa con puntos de referencia y mensajes para ser situados en la ubicación correspondiente.
6. Finalmente, la aplicación muestra al usuario las respectivas alertas de emergencia recibidas.



### 3.6.3. Diagrama de base de datos

El diseño de la base de datos de la aplicación móvil se representa en la figura 25, las tablas utilizadas son empleadas para ingresar y consultar información desde el dispositivo móvil. A continuación, se detalla los campos de cada tabla.

## Diagrama de base de datos

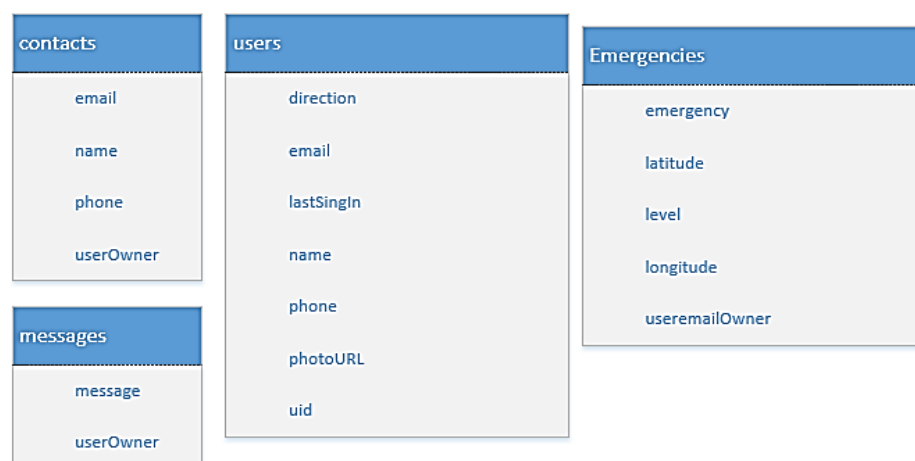


Figura 25 Diagrama base de datos del sistema de botón de pánico.  
Elaborado por: Gabriela Guamán y Freddy Orquera

Tabla 29 Descripción de campos

Tabla	Campo	Descripción
<b>Contacts</b>	Email	Correo electrónico donde será enviado la alerta
	Name	Nombre identificador para registro
	Pone	Teléfono convencional o celular
	UserOwner	Id único de registro
<b>Messages</b>	Message	Mensaje personalizado por el usuario, mismos que se utilizaran en las alertas.
	userOwner	Id único de mensaje único
<b>Users</b>	Direction	Dirección del usuario en caso de emergencia
	Email	Correo de usuario, este debe ser válido para poder recibir alertas.
	lastSingIn	Registro del ultimo ingreso en la aplicación
	Name	Nombre de usuario (preferencia personal)
	Pone	Teléfono convencional o celular
	photoURL	Dirección URL donde almacena la fotografía de perfil.
	Uid	Id único de usuario

<b>Emergencies</b>	Emergency	Mensaje de alerta seleccionado
	Latitude	Coordenadas angulares (norte o sur)
	level	Tipo de alerta según semaforización
	Longitude	Coordenadas angulares (este u oeste)
	useremailOwner	Correo eléctrico de contactos a los que se envió la alerta.

Nota: Tabla que detalla el uso de cada campo de las tablas de la base de datos.

#### 3.6.4. Interfaces

Las pantallas finales del prototipo fue el resultado de varios diseños propuestos, el proceso del diseño final se fue destinado para dispositivos con sistema iOS y Android. A continuación, se detalla cada una de ellas.

- **Iniciar sesión**

La Figura 27 muestra el prototipo de la pantalla principal con dos campos los cuales darán el acceso completo a la aplicación. El primer campo será el correo registrado mientras que el segundo se colocará la contraseña exacta a la que se ingresó en el registro, cualquier error o equivocación no brindará un acceso correcto.

### Interfaz inicio de sesión “SAFE”



Figura 26 Pantalla principal de la aplicación.  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Registrar usuario**

La figura 27 muestra la pantalla de registro de nuevo usuario, donde se despliega campos importantes que se deben completar para un vínculo entre la aplicación y el usuario. La información debe ser auténtica, ya que al ocurrir alguna emergencia el receptor debe recibir la información de manera clara y concisa.

## Interfaz registro de usuario “SAFE”



Figura 27 Pantalla de registro de usuario nuevo.  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Perfil de Usuario**

Esta pantalla muestra los datos del usuario registrado junto con la fotografía, como se puede observar en la figura 28. Adicionalmente, contiene dos botones que permiten la función de editar y salir de la aplicación.

### Interfaz perfil de usuario “SAFE”

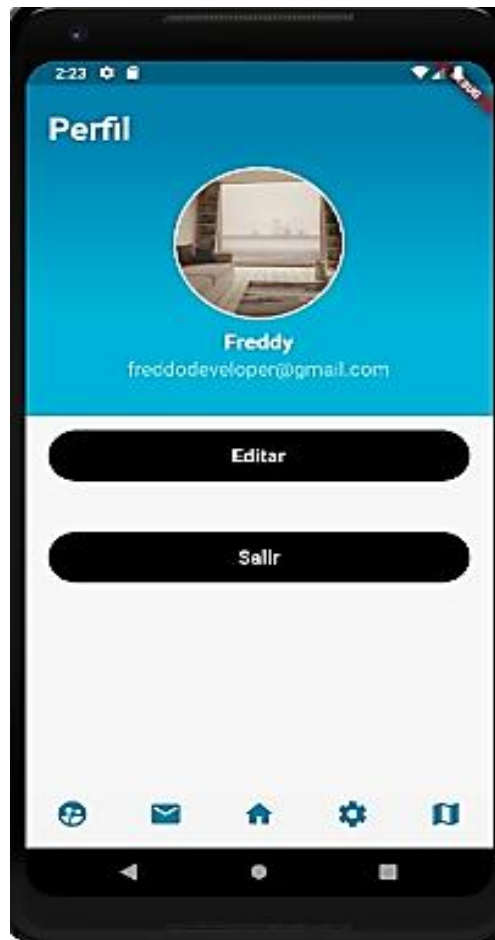


Figura 28 Pantalla de perfil de usuario.

Elaborado por: Gabriela Guamán y Freddy Orquera

- **Editar Perfil**

Esta pantalla muestra la información del usuario en cada campo respectivo, los datos pueden ser editados de manera constante debido a que no existe restricciones o límite de modificaciones, en la figura 29 se observa un ejemplo de datos ingresados de manera correcta en cada campo.



### Interfaz edición de datos “SAFE”

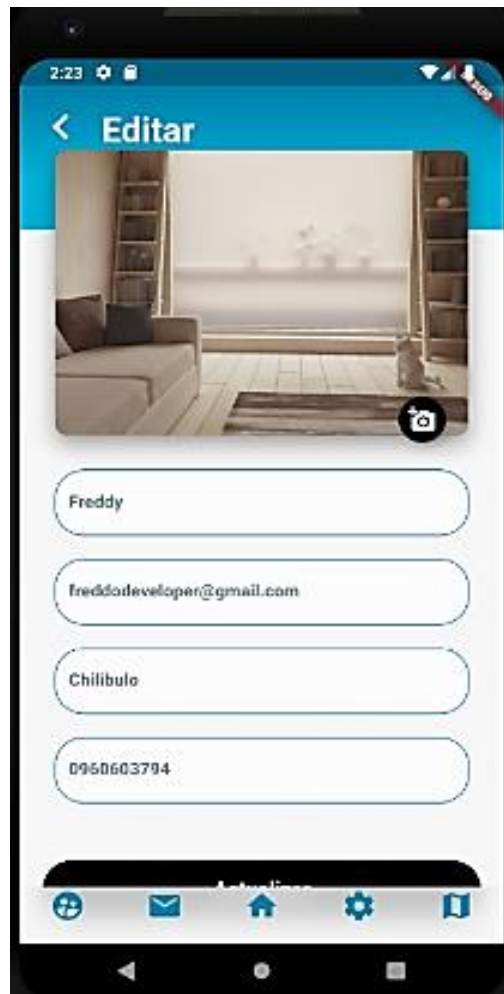


Figura 29 Pantalla de editar datos de usuario.  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Contactos**

La interfaz de contactos se muestra en la figura 30, esta pantalla es indispensable para el envío de alertas desde la aplicación, dado que, al no ser registrado ningún contacto el llamado de emergencia no tendrá un destinatario por lo consecuente no existirá un apoyo al llamado.

### Interfaz edición de contactos “SAFE”



Figura 30 Pantalla de visualización de contactos registrados.  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Registrar Contacto**

La figura 31 representa la interfaz de nuevo registro donde se podrá almacenar un número ilimitado de contactos, sin embargo, se debe tener en cuenta que al poseer un número excesivo de contacto en el momento del envío de alerta puede ocasionar confusión y no terminar el proceso de llamado de emergencia.

### Interfaz de registro de contacto “SAFE”



Figura 31 Pantalla de registro de contacto.  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Semaforización**

Como se ha mencionado con anterioridad la aplicación posee 3 tipos de niveles como muestra la figura 32, cada una posee alertas que son identificadas con colores preseleccionados según su interpretación. Además, poseen una pantalla subsecuente de envío de mensaje con diferente estructura.

### Interfaz de semaforización “SAFE”



Figura 32 Pantalla que muestra los niveles de emergencia.  
Elaborado por: Gabriela Guamán y Freddy Orquera

### **Botón Rojo**

La figura 33 representa la acción de un usuario al presionar el botón rojo para manifestar que se encuentra en un nivel de emergencia muy alto, por lo que solo se puede seleccionar un mensaje de la lista pues los destinatarios son predeterminados para esta alerta.

### Interfaz emergencia Roja “SAFE”

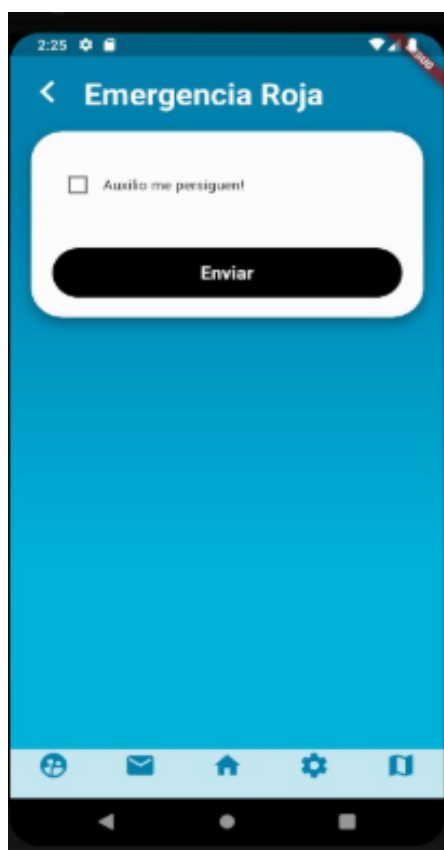


Figura 33 Interfaz nivel rojo.  
Elaborado por: Gabriela Guamán y Freddy Orquera

### Botón Amarillo

La pantalla de botón Amarillo se la puede interpretar como un nivel medio donde la emergencia es relativamente controlable, esta posee la característica de poder seleccionar un mensaje personalizado creado con anterioridad como se observa en la figura 34.

### Interfaz emergencia amarilla “SAFE”



Figura 34 Interfaz nivel amarillo.  
Elaborado por: Gabriela Guamán y Freddy Orquera

### Botón Verde

La pantalla de emergencia verde se la interpreta como un nivel bajo el cual permite personalizar en el momento el mensaje a enviar. Se debe tener en cuenta que los mensajes deben ser claros y concisos para que el receptor tenga una comprensión correcta del mensaje. En la figura 35 se puede visualizar el campo disponible para el mensaje y a su vez la lista de contactos.

### Interfaz emergencia verde “SAFE”



Figura 35 Interfaz de nivel verde  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Mensajería**

La pantalla de mensajería muestra una lista con los mensajes creados, en el caso de iniciar por primera vez la mensajería se encontrará vacía, por lo cual no se podrá enviar una alerta de ningún nivel debido a que no tendrá un mensaje claro y conciso del llamado de emergencia.

Al igual que en los contactos se pueden agregar mensajes sin límite, teniendo en cuenta que es recomendable tener solo los mensajes necesarios y de rápida comprensión para el receptor como se muestra en la figura 36.

## Interfaz mensajería “SAFE”



Figura 36 Interfaz de mensajería  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Nuevo mensaje**

Los mensajes creados deben poseer un contexto claro, se pueden ingresar caracteres como signos de interrogación o exclamación para dar a entender una expresión del momento. El mensaje no debe ser demasiado extenso o sin sentido pues, el destinatario podría no entenderlo la figura 37 muestra la pantalla del prototipo.



### Interfaz nuevo mensaje “SAFE”



Figura 37 Interfaz de nuevo mensaje  
Elaborado por: Gabriela Guamán y Freddy Orquera

- **Mapa en tiempo real**

La figura 38 muestra el mapa en tiempo real de la ubicación del usuario, los mensajes escogidos junto a la ubicación se enviarán al contacto seleccionado. Se debe considerar que para el envío de este se debe tener activado el GPS en el dispositivo.

### Interfaz de mapa “SAFE”



Figura 38 Interfaz de mapa y alertas de emergencia.  
Elaborado por: Gabriela Guamán y Freddy Orquera

### 3.6.5. Diagrama de Clases

En la figura 39 se muestra el diagrama con las clases principales que se intervienen en la aplicación móvil.

Tabla 30 Descripción de clases de aplicación móvil

Clase	Descripción	Función
<b>Iniciar_sesion</b>	Ingreso a la aplicación	Permite el ingreso al sistema mediante credenciales específicas para su autenticación.
<b>cerrar_sesion</b>	Salida de la aplicación	Realiza el cierre de sesión del sistema.

<b>registrarse()</b>	Registrar usuario	Permite registrar los datos de un nuevo usuario.
<b>Editar_perfil</b>	Editar perfil	Accede a la información del usuario para poder ser modificada con los nuevos cambios realizados.
<b>Agregar_contacto</b>	Añadir contacto	Ingreso de datos para nuevos contactos.
<b>Agregar_mensaje</b>	Añadir mensaje	Almacenamiento de mensajes personalizados.
<b>Emergencia_verde</b>	Alerta mínima	Envío de alerta a uno o varios contactos junto con un mensaje personalizado y localización real.
<b>Emergencia_amarilla</b>	Alerta intermedia	Envío de alerta a uno o varios contactos junto con mensaje y localización real.
<b>Emergencia_roja</b>	Alerta grave	Envío de alerta a todos los contactos junto con mensaje y localización real.
<b>Mostrar_alertas_mapa</b>	Historial de alertas	Consulta de puntos en el mapa que muestran las alertas recibidas.

Nota: Esta tabla detalla las clases que contiene la aplicación móvil.

### Diagrama de Clases

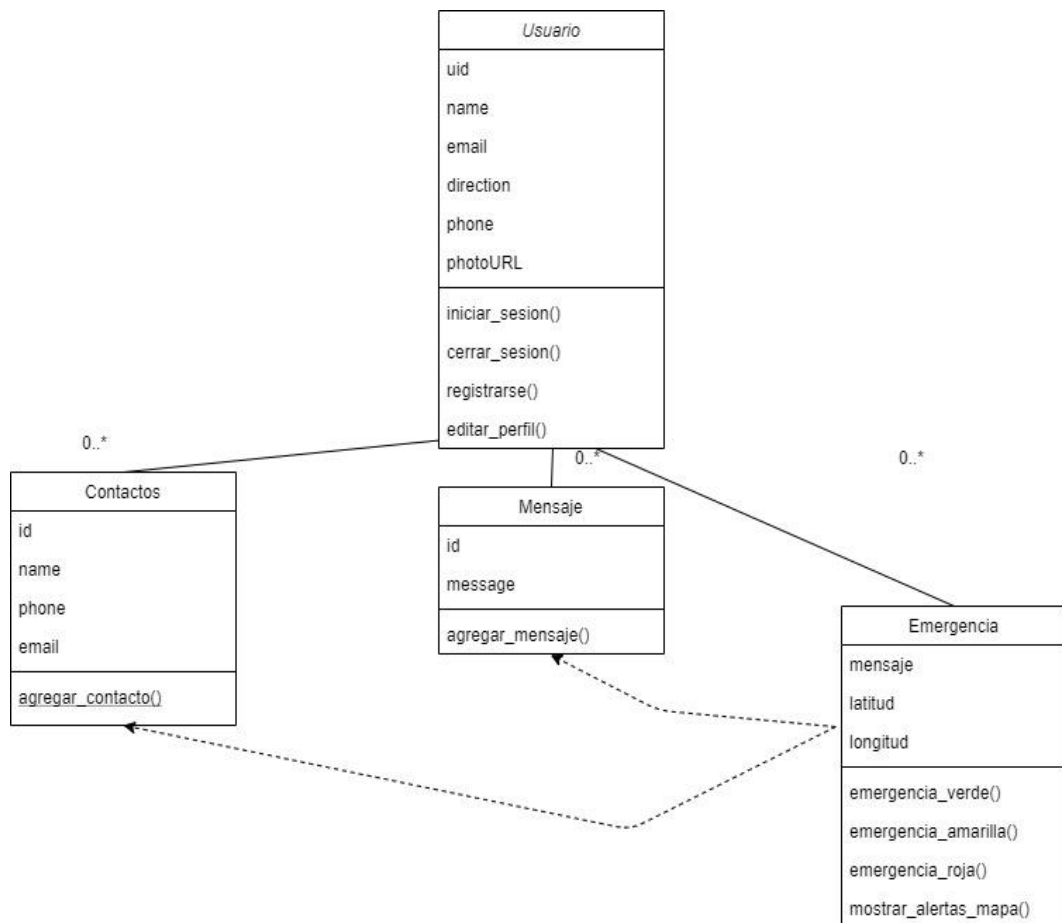


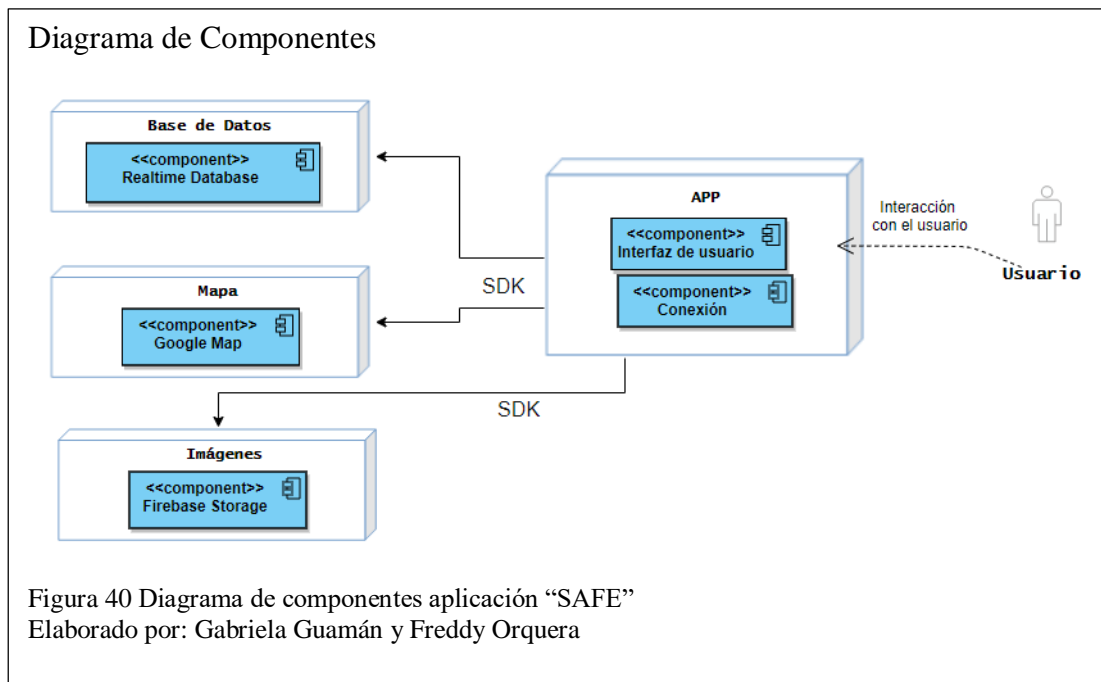
Figura 39 Diagrama de clase aplicación “SAFE”  
Elaborado por: Gabriela Guamán y Freddy Orquera-

### 3.6.6. Diagrama de Componentes

En el diagrama de componentes plasmado en la figura 40 se visualiza una estructura de alto nivel compuesta por una base de datos, mapa y un dispositivo móvil, los cuales se comunican para dar una interacción viable con el usuario.

El usuario interactúa con la aplicación mediante un dispositivo móvil en donde, se ejecuta el aplicativo y se muestra una interfaz adaptable a las necesidades del consumidor. La conexión entre el dispositivo y los componentes permiten realizar las transacciones de almacenamiento y recuperación de información con la base de datos, google maps y firebase storage, esta gestión de información puede ser

incluso cuando el dispositivo se encuentra desconectado, teniendo en cuenta que la sincronización será automática al momento de conectarse a una red.



## Capítulo 4

En el presente capítulo se detallará las distintas fases que intervienen para la conexión entre la base de datos y la aplicación.

### 4.1. Implementación

Este punto detalla una guía de pasos a seguir para la implementación del prototipo, donde se conectará con la herramienta Firebase para almacenar la información.

Es importante recordar que un requisito indispensable antes de crear la conexión con la base de datos es crear y diseñar un proyecto en Android Studio, pues este permitirá la interacción con el usuario y la base de datos.

#### 4.1.1. Creación de proyecto Firebase

Este punto es indispensable para la conexión, por lo que se creará un proyecto con características específicas dependiendo del uso que se le va a dar. A continuación, se describe los pasos a seguir:

- **Otorgar un nombre.**

El nombre del proyecto es indispensable, pues al realizar la documentación o hacer referencia este debe ser un punto de identificación dentro del proyecto. En la figura 41 muestra el ingreso de nombre para el actual proyecto.

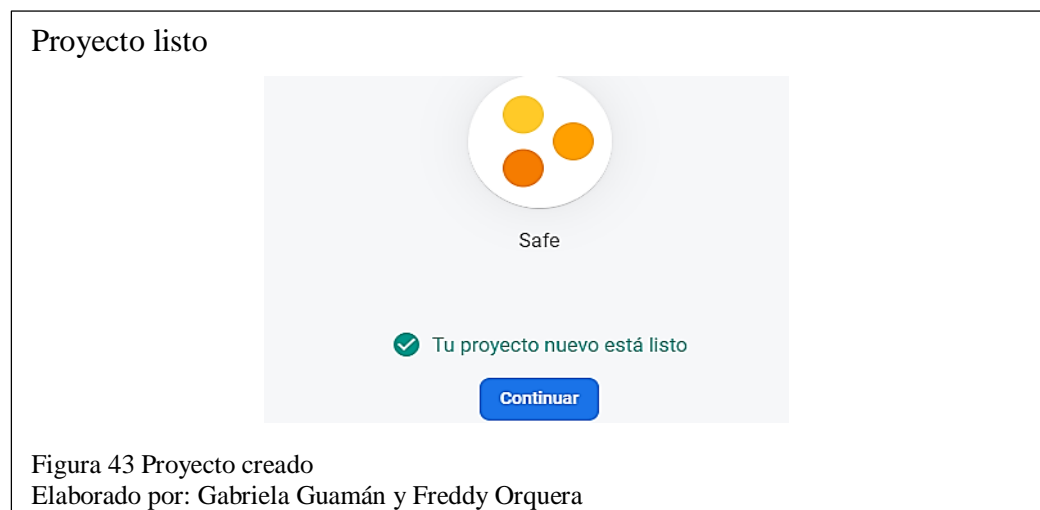


- **Google Analytics**

Esta herramienta podrá ser activada o no, pues brinda la ventaja de poder ser activada posteriormente si fuera necesaria, esta opción ofrece beneficios como el análisis de rendimiento, errores y otras como muestra la figura 42.



Para concluir, como se observa en la figura 43 después de aceptar términos y condiciones se procede a la creación final del proyecto:



#### **4.1.2. Conexión con la base de datos**

La conexión entre un gestor de base de datos externo y la aplicación es una alternativa completamente factible ya sea desde una plataforma Android o iOS.

Por lo cual es necesario realizar la conexión de manera correcta para alcanzar un enlace directo y así no exista pérdida de información. A continuación, se detalla los puntos necesarios para enlazar la aplicación y la base de datos.

- **Registro de proyecto**

Es necesario agregar las plataformas que serán utilizadas con la aplicación estas pueden ser: iOS, Android y web. Cabe recalcar que la guía de registro de iOS y Android son similares.

La figura 44 muestra 3 campos: el primer campo corresponde al nombre del paquete, el cual representa un valor al “applicationId” que se encuentra en el archivo “build\_gradle”, el segundo campo es opcional pues especifica un sobrenombre a la aplicación como representación, este no puede ser visto por los usuarios.

Para finalizar, el tercer campo pertenece a la certificación de firma SHA-1 al igual que el segundo punto es opcional, pues este certificado permite instalar la aplicación en un dispositivo real para empezar a realizar pruebas sin necesidad de generar un APK firmado.



## Registro de proyecto

Figura 44 Registro de app en Firebase

Elaborado por: Gabriela Guamán y Freddy Orquera

- **Archivo de configuraciones**

Firebase generará un documento “google-services.json”, el cual contiene una serie de configuraciones que permiten establecer la conexión con las aplicaciones por lo que no sufrirá ninguna modificación.

Como indica la figura 45 este archivo será copiado en el módulo de app.

## Google service

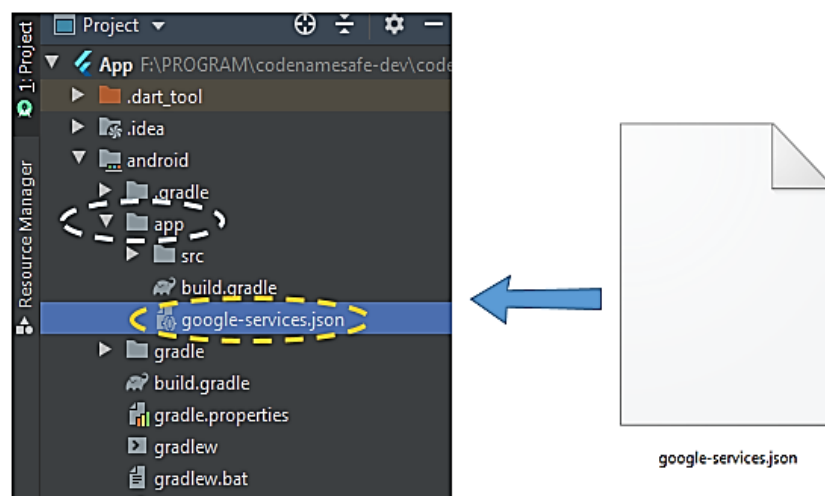
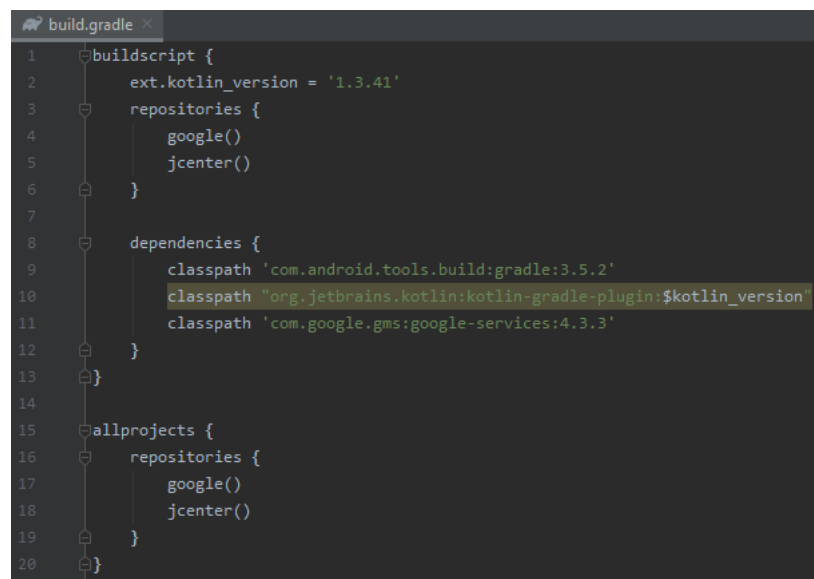


Figura 45 Archivo de configuraciones copiado en app.

Elaborado por: Gabriela Guamán y Freddy Orquera

- **SDK de Firebase :** El archivo “google-services.json contiene build.gradle, un complemento que será modificado con una estructura e información relevante para su uso, para lo cual la modificación se realizara de forma rápida con la copia de varias líneas de código predefinidas a nuestra aplicación como se indica en la figura 46.

Archivo build.gradle



```
1 buildscript {
2     ext.kotlin_version = '1.3.41'
3     repositories {
4         google()
5         jcenter()
6     }
7
8     dependencies {
9         classpath 'com.android.tools.build:gradle:3.5.2'
10        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
11        classpath 'com.google.gms:google-services:4.3.3'
12    }
13 }
14
15 allprojects {
16     repositories {
17         google()
18         jcenter()
19     }
20 }
```

Figura 46 Archivo build.gradle modificado y sincronizado.  
Elaborado por: Gabriela Guamán y Freddy Orquera

Una vez finalizado la configuración de servicios Firebase la aplicación podrá empezar a interactuar con la base de datos.

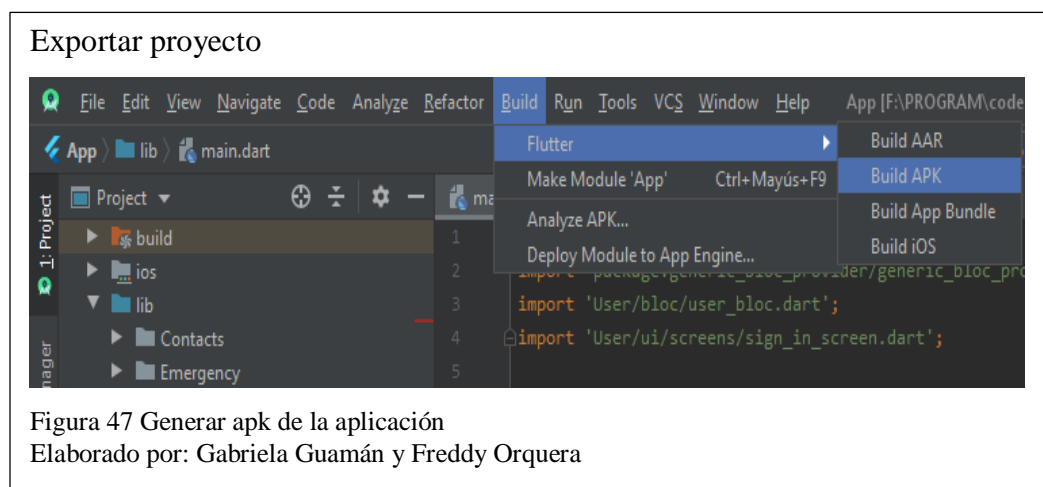
## 4.2. Implantación

Una vez verificado los componentes y terminada la fase de desarrollo se procede a ejecutar la implantación de la aplicación en un dispositivo móvil, este punto abarca el detalle de la compilación e instalación.

### 4.2.1. Exportar apk

El instalador es un empaquetado de datos indispensable que permite la instalación de componentes necesarios para la ejecución de la aplicación en un sistema operativo. Para la exportación del instalador se debe seguir los siguientes pasos:

- En primer lugar, de la barra superior de herramientas se selecciona la opción “Build” donde se despliega un submenú en el que se escoge “Flutter” este a su vez presentar un menú de empaquetado. Se debe seleccionar “Build APK” como muestra la figura 47.
- Para finalizar, el ejecutable que será distribuido a los usuarios se encuentra generado en la siguiente ruta: “F: \nombreproyecto\ App\ build\ app\outputs\apk\release”.



#### 4.2.2. Instalación

Los pasos para instalar la aplicación en un dispositivo móvil serán igual para todas las plataformas, pues no requiere de alguna configuración adicional solo las que se especifican a continuación:

- La aplicación podrá ser descargada, conectada con cable usb a la computadora y colocada en a ubicación interna del dispositivo a elección del usuario o simplemente copiar en una tarjeta SD. En la figura 48 muestra el archivo “.apk” y en este caso el ejecutable se encuentra descargado desde el internet.

## Obtención de aplicación

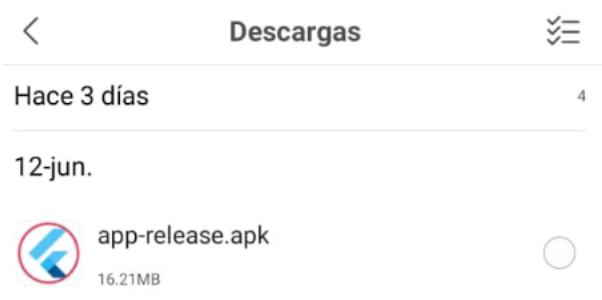


Figura 48 Ubicación del apk en el dispositivo móvil  
Elaborado por: Gabriela Guamán y Freddy Orquera

- Una vez presionado la aplicación para su ejecución en el dispositivo, esta muestra un mensaje de seguridad que permite informar una vez más al usuario que la aplicación no es creada o propia del dispositivo. En el mensaje se debe ingresar a la configuración para poder dar permiso de instalación de fuentes desconocidas como muestra la figura 49.

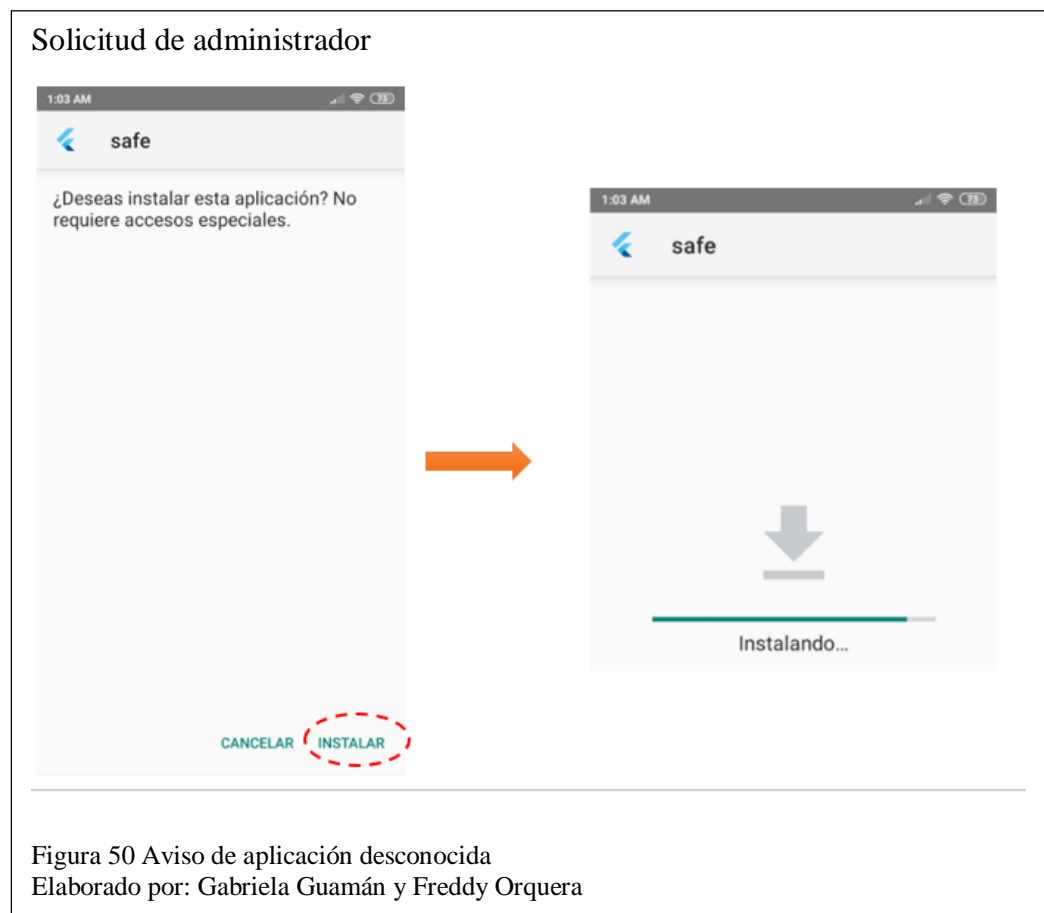
## Solicitud de administrador



Figura 49 Aviso de aplicación desconocida  
Elaborado por: Gabriela Guamán y Freddy Orquera

Finalmente, el dispositivo por seguridad muestra nuevamente un mensaje de advertencia donde indica el nombre de la aplicación que va a ser instalada dando al usuario la potestad de aceptar o rechazar la instalación. Una vez presionado “Instalar” como muestra la figura 50 el apk se ejecutará dando paso a la descomposición de datos necesarios para una instalación exitosa en el dispositivo.

Una vez finalizado el proceso de instalación, el usuario podrá acceder y hacer uso de la aplicación como cualquier otra, pues está podrá ser utilizada en cualquier momento



## Capítulo 5

En el siguiente capítulo se realizará pruebas de funcionamiento de la aplicación, donde se establecerá diversas métricas con el objetivo de identificar y mejorar la calidad del producto.

### 5.1. Métricas de Evaluación

Las métricas son una escala de unidad que se establece para obtener una medición en las diferentes pruebas de la aplicación, estas serán usadas como indicadores para comprobar el cumplimiento de los requisitos planteados por el usuario.

#### 5.1.1. Lista de métricas

A continuación, se lista las métricas que se evaluarán a la aplicación para dar un valor cuantitativo en el análisis.

- **Fallos de aplicación:** Número de fallos críticos que ocasionan un cierre inesperado de la aplicación.
- **Uso del CPU:** Porcentaje de uso del CPU al ejecutar la aplicación.
- **Latencia:** Tiempo promedio de respuesta que produce una interacción en la aplicación.

#### 5.1.2. Esquema de puntuación

Una vez definido las métricas, se procede a establecer la escala de puntuación para cada una de ellas. Dependiendo la especificación se colocará el rango o valor con el que se va medir ese indicador.

Tabla 31 Valores de medición

Métrica	Bueno	Aceptable	No aceptable
<b>Fallos de aplicación</b>	0	1 – 2	3 en adelante
<b>Uso del CPU</b>	1% - 20%	21% - 70%	71% en adelante
<b>Latencia</b>	1ms - 400ms	401ms - 2000ms	2001ms en adelante

Nota: Rangos establecidos para cada métrica.

Con la finalidad de obtener datos reales, la información obtenida se evaluará con los valores indicados en la Tabla 30, por ende, las medidas también serán diferentes para cada métrica, pues se tiene datos numéricos, porcentaje y milisegundo. Para cada rango se especificó un valor cualitativo que lo representa como se describe a continuación:

- **Bueno:** Representa un funcionamiento eficiente que no necesita cambios o ajustes.
- **Aceptable:** Simboliza un funcionamiento intermedio, por lo que no es obligatorio realizar mejoras.
- **No aceptable:** Representa un funcionamiento erróneo o fallido, por lo cual es indispensable realizar mejoras o cambios.

## 5.2. Pruebas

Las pruebas de la aplicación representan un punto importante, pues este garantiza un funcionamiento adecuado al usuario. Las pruebas que se realizaran a la aplicación son: pruebas de sistema y de funcionamiento, estas se especifican a continuación.

### 5.2.1. Pruebas de funcionamiento

En este punto, se realizará un análisis completo a la aplicación para la obtención de resultados de dichas pruebas. Adicionalmente se comprobará la conexión que posee la aplicación con la base de datos, pues este apartado es fundamental para comprobar si no existe desvío o pérdida de información

A continuación, se detalla la prueba de funcionamiento de cada interfaz.

- **Interfaz registro**

Esta prueba tiene como objetivo comprobar que la información solicitada en el formulario de registro se almacene en la base de datos incluyendo la encriptación de la contraseña. El campo contraseña debe reemplazar cada letra ingresada por un carácter especial ‘●’.

Prueba de registro

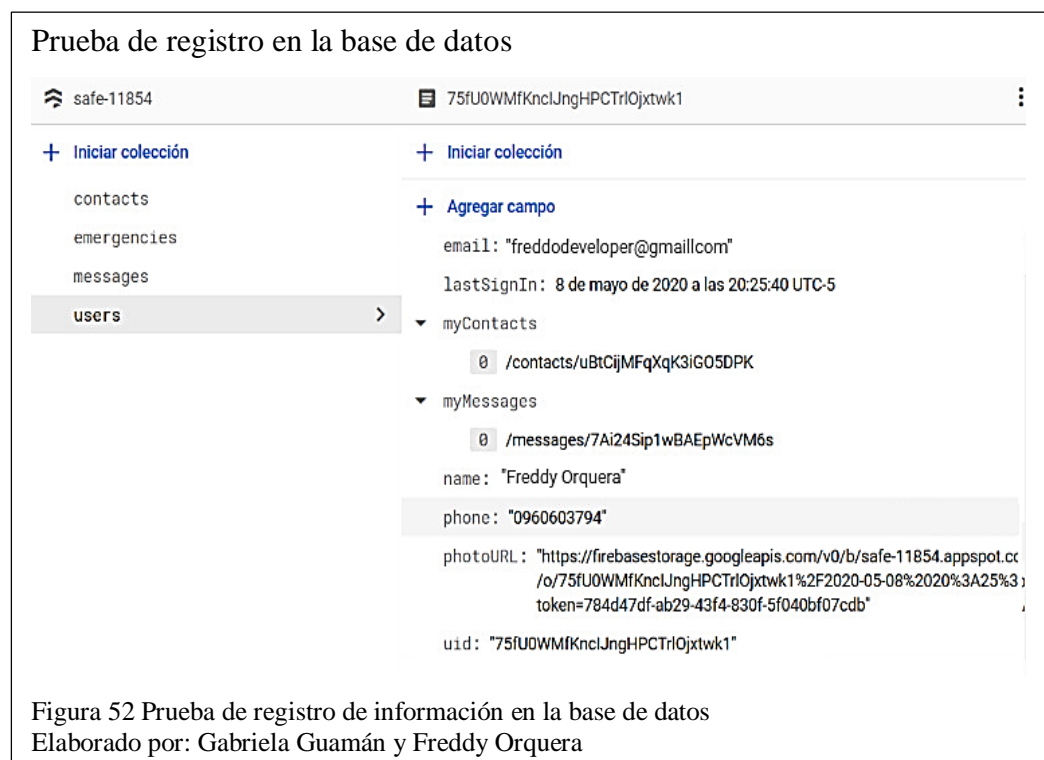


Figura 51 Prueba de registro en aplicación SAFE  
Elaborado por: Gabriela Guamán y Freddy Orquera



La figura 51 muestra el ingreso de información para un usuario nuevo, se puede apreciar que en el campo de contraseña cada carácter es reemplazado por la viñeta antes mencionada protegiendo de este modo la privacidad del usuario. El campo correo se encuentra validado, por lo que no permite ingresar una estructura diferente a [xxxxx@xxxx.xxx](#).

Como se observa en la figura 52 los datos fueron almacenados de manera correcta, por lo cual se puede decir que la prueba del módulo ha sido exitosa.



#### 4.3.1. Prueba de Login

Esta prueba tiene como objetivo comprobar que los usuarios ingresen a la aplicación colocando de manera correcta los campos de usuario y contraseña. Para ello, se realizará dos intentos: el primero se introducirá la clave invalida y correo correcto, mientras que en la segunda se ingresará en ambos campos los datos correctos.

## Prueba de Login

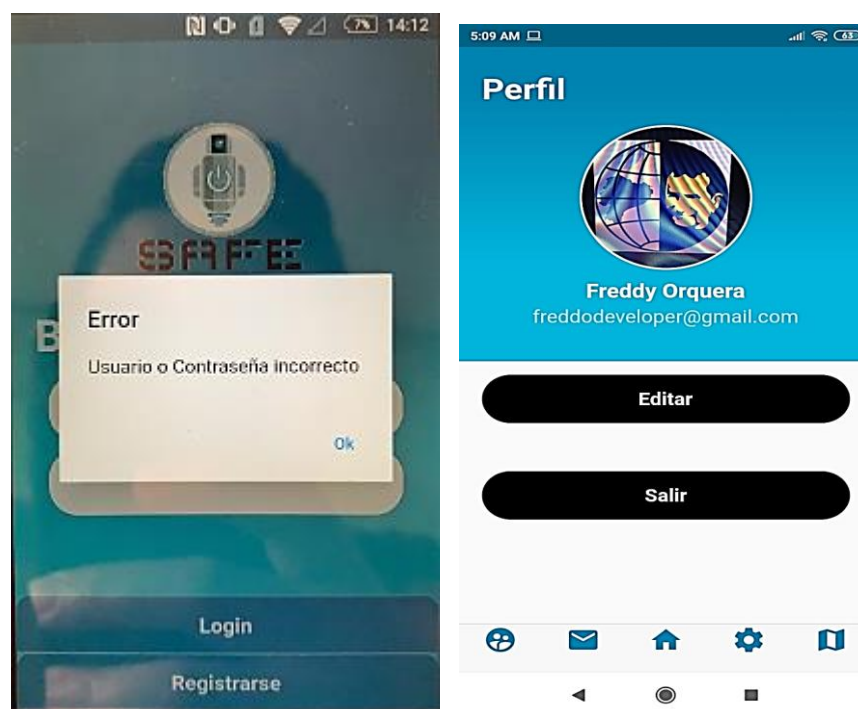


Figura 53 Prueba de login en aplicación SAFE  
Elaborado por: Gabriela Guamán y Freddy Orquera

La figura 53, plasma dos expectativas diferentes: la primera muestra un mensaje de notificación de logueo erróneo, pues de este modo indicando al usuario el ingreso fallido de la autenticación. La segunda muestra una autenticación exitosa, puesto que, en este caso los datos ingresados en los dos campos fueron correctos.

Por tanto, se puede aseverar que la interacción entre base de datos y aplicación es satisfactoria, al igual que las validaciones de campo para usuarios y clave.

- **Interfaz contactos**

Esta prueba tiene como objetivo ingresar un contacto y registrarlo para continuamente comprobar su ingreso en la base de datos. Para lo cual en el transcurso del proceso se comprobará validaciones de cada campo.

### Prueba de registro de contactos

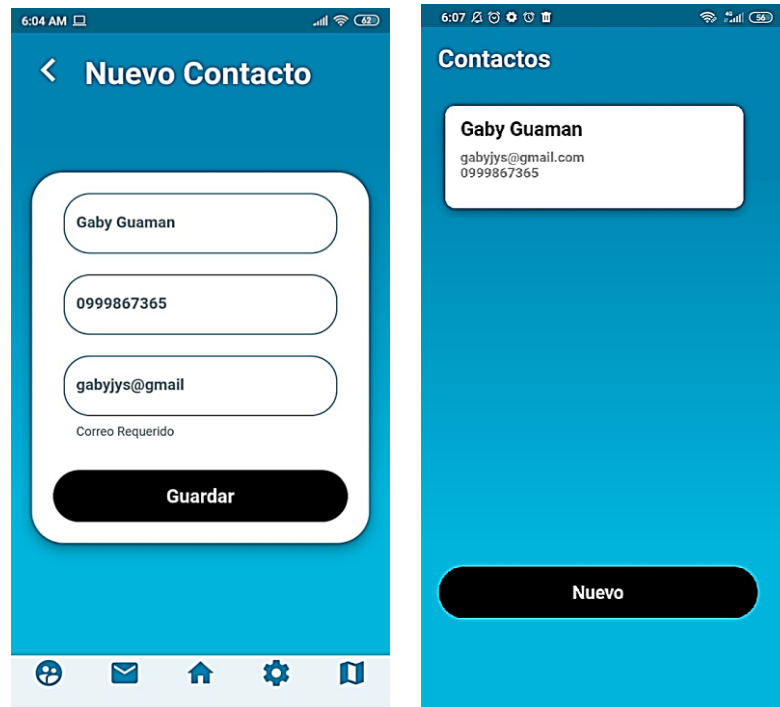


Figura 54 Prueba de contactos en aplicación SAFE  
Elaborado por: Gabriela Guamán y Freddy Orquera

La figura 54 contiene dos imágenes la primera se ingresa los datos del nuevo contacto, se recalca la validación del campo correo ya que no se encuentra completo y por lo tanto la aplicación impide almacenar la información. Para la segunda imagen se observa ya el ingreso del contacto previamente corregido.

### Prueba de registro en la base de datos

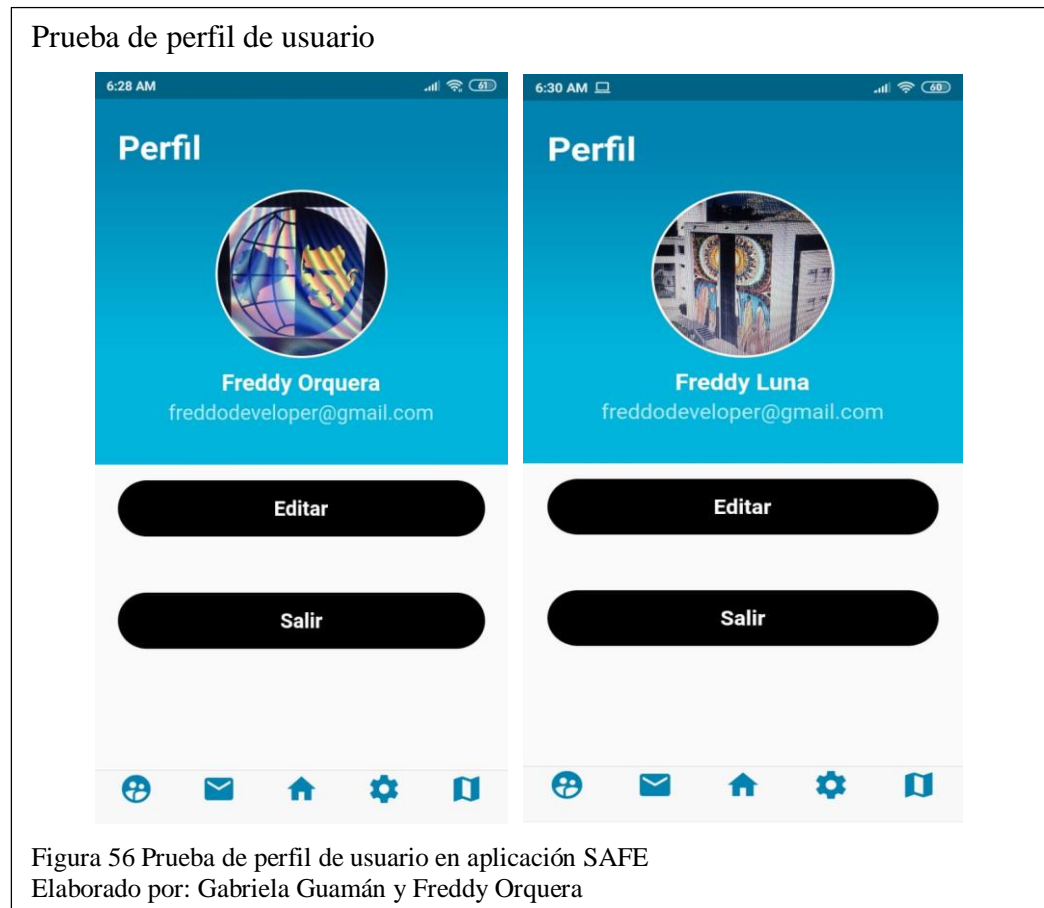
prueba-dac34	contacts	tropL9hlp5oGqj4hiHh9
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
contacts >	tropL9hlp5oGqj4hiHh9 >	+ Agregar campo
emergencies		email: "gabyjys@gmail.com"
messages		name: "Gaby Guaman"
users		phone: "0999867365"
		userOwner: "/user/tropL9hlp5oGqj4hiHh9"

Figura 55 Prueba de revisión de información en la base de datos.  
Elaborado por: Gabriela Guamán y Freddy Orquera

La información ingresada se verifica en la base de datos, de esta manera se puede afirmar que la prueba que se realizó es exitosa.

- **Interfaz modificación de perfil**

Esta prueba consiste en ingresar al perfil personal creado al momento de registrar el usuario y realizar una modificación válida, esta debe ser instantánea.

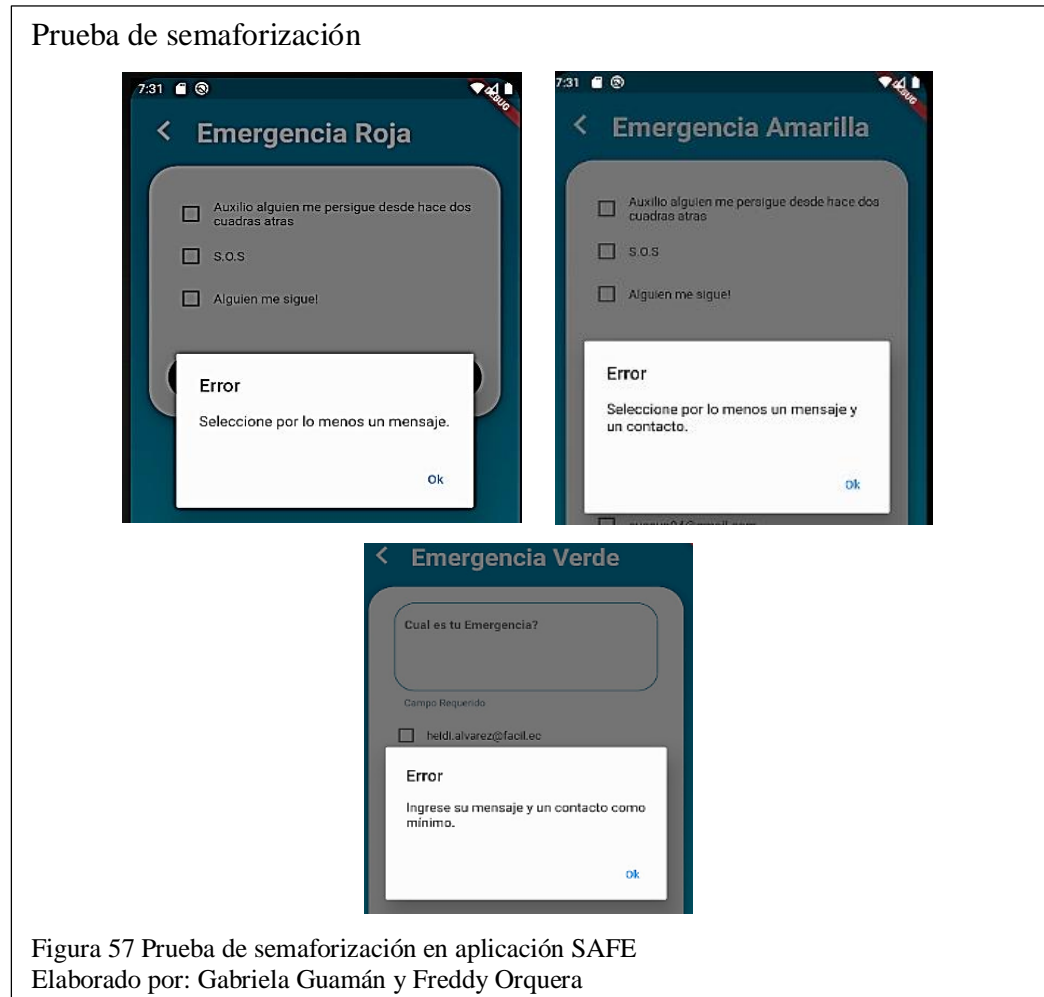


La figura 56 muestra la modificación del campo nombre y la fotografía, como se puede observar el tiempo de actualización fue de 2 minutos. Esto incluye el tpeo, la activación de cámara para la nueva fotografía y el servicio de conexión (internet).

El tiempo de respuesta y el resultado obtenido indica que el proceso se ejecutó de manera correcta, pues no se registró datos erróneos o faltantes. Por ello, se puede afirmar que la prueba fue exitosa.

- **Interfaz semaforización**

Esta prueba tiene como objetivo analizar cada color y las subpantallas que contiene cada uno, por lo que se va a enviar una alerta de cada nivel. Los llamados que se envían pueden ser sin texto, sin contacto o vacío.



La figura 57 muestra la validación que existe en los 3 niveles, en el nivel rojo se intentó enviar una alerta sin un mensaje debido a que los contactos son por defecto todos los registrados, por lo que el mensaje desplegado aclara al usuario que debe escoger por lómenos un mensaje para ser enviado y notificar la alerta al receptor. En el nivel amarillo, la alerta se intentó enviar sin mensaje ni correo electrónico donde inmediatamente se desplegó un mensaje de aviso al usuario en donde

informa que se debe seleccionar mínimo un mensaje y un contacto para que la alerta sea activada.

Para finalizar el color verde, en esta pantalla se intentó enviar a un contacto un mensaje vacío ya que en este nivel se puede escribir a libre criterio el mensaje que se desee. La alerta no fue procesada debido a que no puede ser enviada una ubicación con un mensaje vacío, pues muchas personas podían interpretarlo como una burla y no darle importancia.

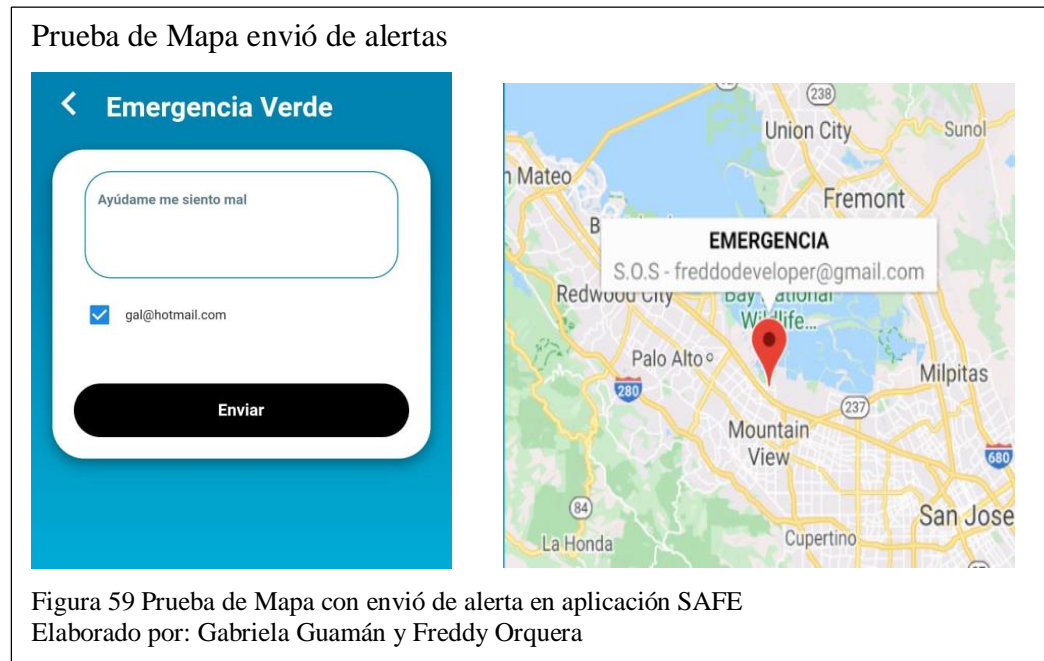
Las pruebas en los 3 niveles fueron exitosas, por lo que se puede afirmar que existe validaciones con mensajes claros y concisos que se presentan al usuario incluso si este desconoce del tema de la tecnología.

- **Interfaz mapa**

Esta prueba tiene como objetivo identificar los puntos de alerta situados dentro del mapa tanto enviados como recibidos. Para esto se procederá a abrir el mapa y presionar un punto de notificación de ayuda para esto el dispositivo se encontrará sin Internet. Mientras que para verificar las alertas enviadas se realizará el mismo proceso en el teléfono del contacto al que se le envió, cabe recalcar que la alerta debe ser enviada con conexión a la red.



La figura 58 muestra el punto de auxilio que se recibió y al presionarlo despliega el mensaje, cabe mencionar que se encuentra sin conexión de red, por lo cual una vez recibida la información está es almacenada en la aplicación y visible para el usuario en cualquier momento.



La figura 59 en cambio se realizó el envío de una emergencia nivel verde, misma que verifica la recepción de esta mediante el mapa del usuario que actuó como destinatario para el auxilio. Esta alerta fue enviada con conexión de red en ambos dispositivos tanto emisor como receptor.

Al haber obtenido las alertas en ambos casos sin problemas tanto en posición del mapa como en mensaje se puede manifestar que la prueba ha sido exitosa.

### 5.2.1. Pruebas de sistema

La prueba se ha realizado de forma automatizada a través de Firebase Test Lab, la cual permite ejecutar la aplicación en distintos dispositivos físicos con distintos niveles de API, cabe aclarar que la aplicación fue diseñada para trabajar con API con versión 21 o superior.

- **Uso del CPU**

La prueba se realizó en dos teléfonos móviles con diferentes características, donde se realizó varias interacciones entre el usuario y la aplicación logrando de este modo obtener el mayor uso de capacidad de CPU del dispositivo.

Las características de estos fueron:

Tabla 32 Características dispositivo 1

<b>Numero de modelo:</b>	<b>ALE-L21</b>
<b>CPU</b>	8 núcleos 1.2 GHz
<b>Memoria</b>	16,00 GB en total
<b>Versión Android</b>	5.0.1

Nota: Detalle de versión del primer dispositivo

Tabla 33 Características dispositivo 2

<b>Numero de modelo:</b>	<b>REDMIN 6A</b>
<b>CPU</b>	Quad-core Max 2.00 GHz
<b>Memoria</b>	32,00 GB en total
<b>Versión Android</b>	9

Nota: Detalle de versión del segundo dispositivo

La prueba fue realizada en el mismo sitio con dos usuarios que no tenían capacitación previa del manejo de la aplicación, los dos dispositivos se encontraban al 100% de batería y sin aplicaciones adicionales funcionando en el momento de la prueba.



- **Latencia**

La prueba que se realizó fue la interacción de tres usuarios que tenían una capacitación previa del funcionamiento de la aplicación, en esta prueba se medirá el tiempo de respuesta de cada interacción entre el dispositivo y el usuario. En la Tabla 34 se detalla el tipo de internet y la edad del usuario.

Tabla 34 Descripción del usuario

Usuario	Edad	Operadora
<b>Lida Guamán</b>	45 años	Claro
<b>Gustavo Caizatoa</b>	54 años	Cnt
<b>Galo Jaramillo</b>	29 años	Netlife

Nota: Detalle de usuarios que participaron como tester.

### 5.3. Resultados

Concluida la etapa de desarrollo e implantación de la aplicación se procede a la evaluación de los resultados de las pruebas realizadas en un dispositivo móvil. La aplicación se encuentra dirigida a todos los ciudadanos a nivel nacional.

- **Comportamiento en dispositivos físicos**

Las pruebas fueron realizadas en diferentes dispositivos con variedad de marca y velocidades de internet, pues al no encontrarse enfocado en un solo grupo de personas es indispensable examinar el comportamiento en todos los perfiles de usuarios que puedan interactuar con la aplicación.

Tabla 35 Resultados de interacción en dispositivos físicos

Dispositivo	API	Fallas (Crash)	Acciones	Duración (min)	CPU	Rendimiento
<b>Pixel 4</b>	29	0	378	17:36	3%	234ms
<b>Xiaomi Mi 5s Plus</b>	26	0	370	20:09	6%	1s 123ms
<b>OnePlus One</b>	22	1	223	20:03	10%	532ms
<b>Galaxy S7 edge</b>	23	0	204	20:05	7%	465ms
<b>Samsung Galaxy Note 8 (SM-N950F)</b>	28	0	30	1:52	7%	719ms
<b>Galaxy J1 ace SM-J111M</b>	22	0	203	10:31	5%	820ms
<b>Pixel 2</b>	28	0	408	20:04	3%	330ms
<b>A9 2018</b>	26	0	367	20:03	4%	388ms
<b>Xiaomi MI 8</b>	28	0	443	20:07	3%	317ms
<b>Sony XPERIA XZ Premium</b>	26	0	454	20:02	3%	292ms
<b>Xiaomi MIX</b>	26	0	382	20:04	7%	345ms
<b>Nexus 5</b>	21	0	20	2:13	2%	622ms

Nota: Los datos son variantes debido a las características que posee cada dispositivo.

La tabla 35 detalla los datos recogidos en cada dispositivo esta prueba fue realizada con la última versión de la aplicación, por lo que al analizar los resultados se obtiene que si un dispositivo realiza interacciones con la aplicación ya sea revisando su perfil, alertas, mensajes, etc., mientras mayor característica posea

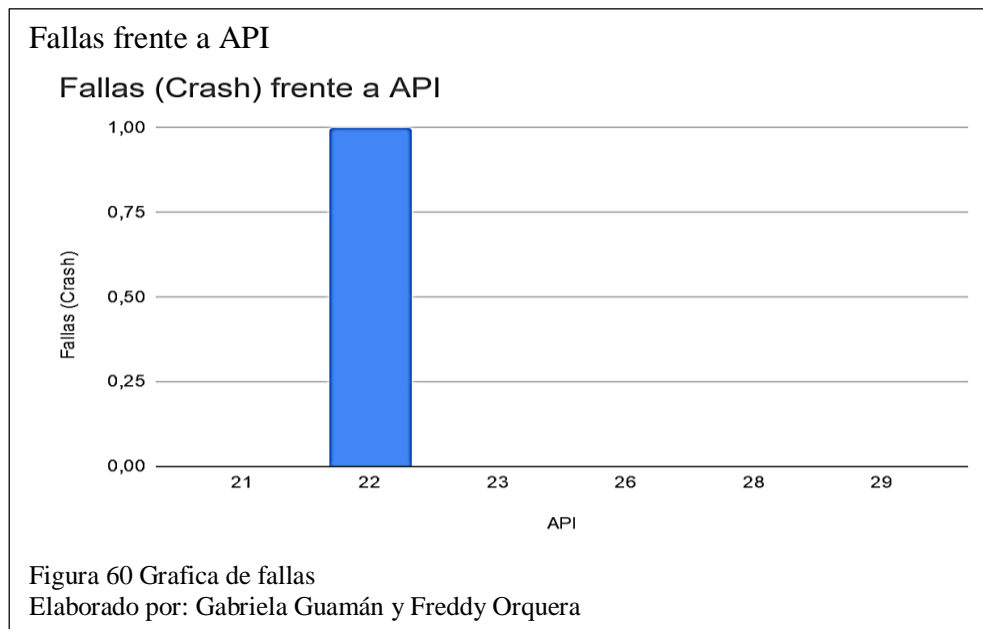
tendrá un alto rendimiento y exactitud. Eso no quiere decir que un dispositivo de menor gama va a tener mayor complicación con la aplicación, sin embargo, al poseer sistemas operativos antiguos o características bajas la precisión del mapa y el rendimiento irá variando.

- **Fallos de aplicación**

Tabla 36 Crash en versiones

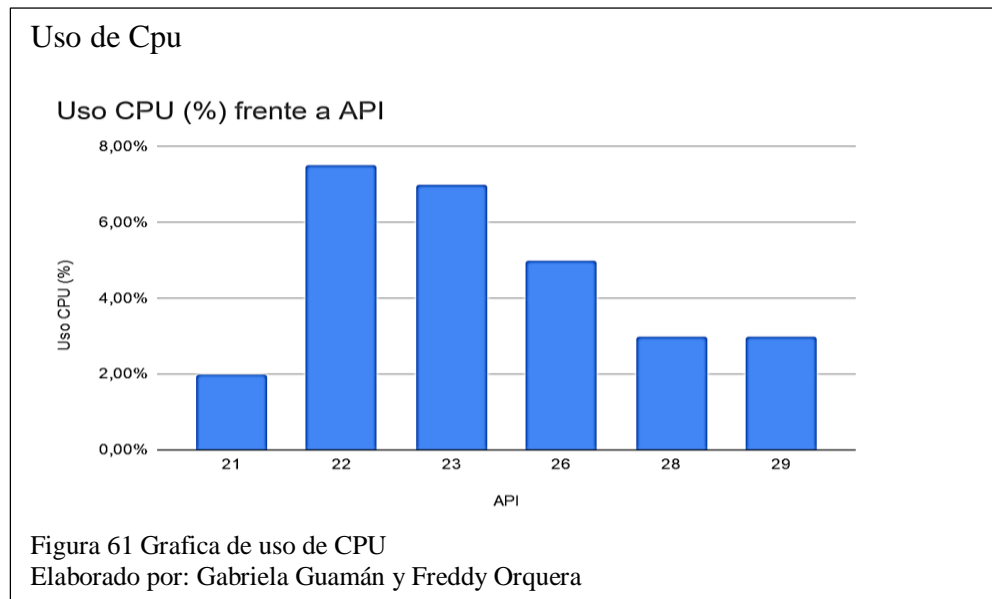
Total fallos	1
APIs	6

Nota: Datos obtenidos con diferentes teléfonos y características.



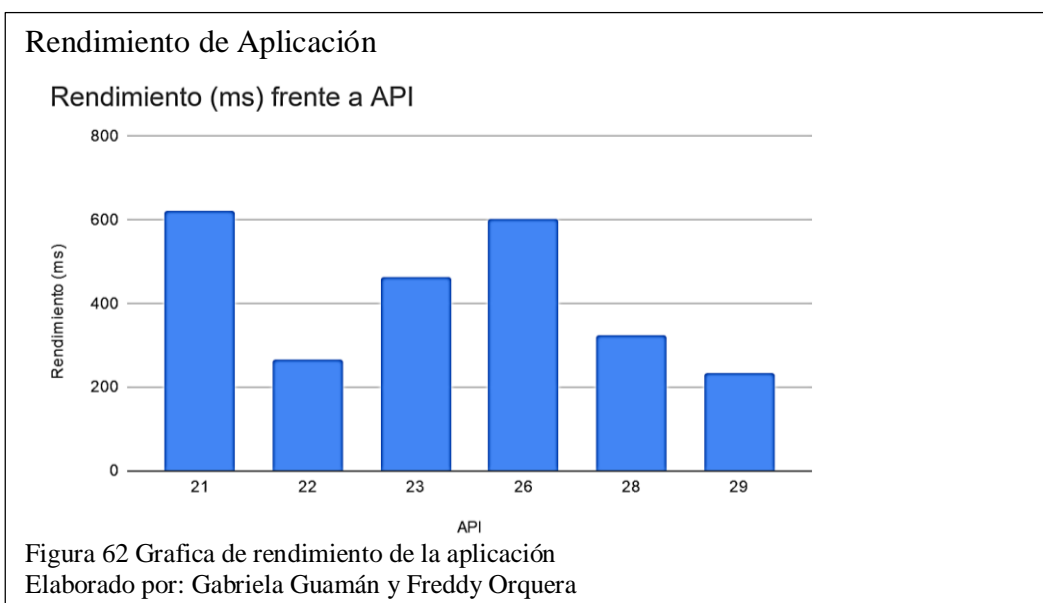
La figura 60 muestra la gráfica de las fallas en dispositivos físicos, donde arroja un total de 1 fallo crítico ocasionando que la aplicación se cierre inesperadamente en la API con versión 22, el error que ocasiono la falla fue la funcionalidad del teclado debido a que al ingresar en un campo el teclado se despliega de manera automática de tal forma que si se presiona fuera del campo sucesivamente genera un cierre de teclado, por lo que provoco en este caso un colapso en la aplicación y se cerró. Sin embargo, el tango de métricas de fallos recae en la sección de “Aceptable”.

- **Uso de CPU**



Al desarrollar una aplicación móvil híbrida uno de los puntos más esenciales es el uso del CPU, al no usar elementos nativos ya sea de Android o iOS puede ser una sobrecarga de procesos el hecho de ejecutar la aplicación, las optimizaciones llevadas a cabo han dado como resultado un margen de uso máximo del 7.5% del CPU, como se observa en la figura 61 mientras más actual sea la versión tecnología del dispositivo menor es consumo del CPU.

- **Latencia**



En la figura 62 se puede apreciar los tiempos de respuesta de la aplicación, dichos tiempos son los de carga de pantalla que se producen al ingresar un texto y ejecutar una acción ya sea agregar, modificar, entre otros. Los datos varían dependiendo la conexión con la base de datos pues para la prueba se lo realizó con operadoras diferentes logrando obtener tiempos más reales, sin embargo, al analizar y comparar los resultados con la tabla de valores establecidos para esta métrica los datos dan como resolución una prueba “Óptima” y “Aceptable”, esto se debe a las características de la aplicación, pues al manejar datos en tiempo real la información es procesada y cargada al instante.

## CONCLUSIONES

- El desarrollo y las pruebas realizadas han sido exitosas por lo cual se ha cumplido con los objetivos planteados, adicional a esto cabe recalcar que los tiempos de respuesta son no perceptible para el usuario, es decir la comunicación es instantánea, cabe acotar que mientras más actual sea la tecnología menor será el tiempo de respuesta.
- La interfaz de la aplicación fue diseñada de tal forma que se adapte a la gran variedad de dispositivos móviles y que sea soportada tanto por Android como iOS, se estableció los tamaños de botones con la finalidad de brindar un acceso al usuario. Adicionalmente los procesos no requieren más de tres pulsaciones que es lo óptimo.
- La aplicación será libre por lo que cualquier persona que tengan acceso a un dispositivo móvil podrá instalarla y así comunicar de manera eficaz una emergencia o inconveniente en el que pueda encontrarse el usuario en ese momento.

## **RECOMENDACIONES**

- Se recomienda realizar un análisis presupuestario para adquirir el certificado de Apple Developer, el cual permita realizar pruebas de optimización en iPhones físicos.
- Para futuras versiones se puede implementar la notificación a correo electrónico o número de teléfono logrando de esta manera brindar al usuario un aviso múltiple.
- Para nuevas implementaciones o mejoras a esta aplicación es indispensable analizar la situación actual tanto en tecnología como en versión de sistemas operativos móviles, pues es importante poseer siempre una adaptabilidad a todos los dispositivos.

## LISTAS DE REFERENCIAS

- Angulo, M. (2018, mayo 7). La Magia de Flutter. Recuperado 4 de marzo de 2020, de SlideShare website: <https://es.slideshare.net/mauricio-angulo/la-magia-de-flutter>
- Apple Inc. (2019). Seguridad de iOS. *Apple*, 111. Recuperado de [https://www.apple.com/es/business/docs/site/iOS\\_Security\\_Guide.pdf](https://www.apple.com/es/business/docs/site/iOS_Security_Guide.pdf)
- Bravo, D. (2019). Un grupo contra la inseguridad en el norte de Quito. Recuperado 20 de noviembre de 2019, de <https://www.ultimasnoticias.ec/las-ultimas/colectivo-inseguridad-quito-robos-vecinos.html>
- Castelo Rivas, W. P., Castelo Rivas, A. F., & Rodríguez Díaz, J. L. (2016). Satisfacción de pacientes con atención médica en emergencias. *Revista Cubana de Enfermería*, 32(3), 335-342. <https://doi.org/10.20538/1682-0363-2017-1-152-161>
- Domínguez Mateos, F., Paredes Velasco, M., & Santacruz Valencia, L. P. (2014). *Programación Multimedia y Dispositivos Móviles*. Recuperado de [https://books.google.com.ec/books?id=gIfDwAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=gIfDwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- EL UNIVERSO. (2020, febrero). Con 600 botones de pánico se lucha contra maltrato a la mujer en Guayaquil . Recuperado 5 de marzo de 2020, de EL UNIVERSO website: <https://www.eluniverso.com/guayaquil/2020/02/20/nota/7747758/guayaquil-violencia-mujer-boton-auxilio-policia-amiga-ya-no-estas>
- Felici, S. (2019). *Sistemas Operativos*. Recuperado de <https://informatica.uv.es/it3guia/FT/cap5-ssoo-ft.pdf>
- Figuerola, L. del C. (2009). *CAPÍTULO II CONCEPTOS Y TIPOS DE DELINCUENCIA*.
- Francia, J. (2017). El Daily Scrum No es una reunión de estado. Recuperado 31 de marzo de 2020, de Scrum.org website: <https://www.scrum.org/resources/blog/el-daily-scrum-no-es-una-reunion-de-estado>
- Fundació Jesuïtes Educació, U. O. de C. (2019). Desarrollo de aplicaciones híbridas, ventajas e inconvenientes. Recuperado 4 de marzo de 2020, de Fundació Jesuïtes Educació website: <https://fp.uoc.fje.edu/blog/desarrollo-de-aplicaciones-hibridas-ventajas-e-inconvenientes/>
- Gachet, I., & Rosero, J. (2015). *Evaluación de Impacto Sistema ECU 911*.
- García Ocanto, S., Barbosa, C., Salinas, M. E., Cousinet, G., Georgi, S., Puebla, A., ... Diblasi, G. (2003). Inseguridad–Seguridad Conceptualización. *Plan de Seguridad UNCuyo*, 125. Recuperado de [http://www.uncuyo.edu.ar/relaciones\\_institucionales/upload/documento-completoplan-de-seguridad1.pdf](http://www.uncuyo.edu.ar/relaciones_institucionales/upload/documento-completoplan-de-seguridad1.pdf)
- Gil Villa, F. (2019). La función social punitiva en Iberoamérica.Circunstancias globales y locales. *Revista Latinoamericana de Estudios de Seguridad*, 24, 10-25.



- Hernandez, U. (2019, mayo 13). El rey multiplataforma: Flutter. Recuperado 4 de marzo de 2020, de [códigofacilito website: https://codigofacilito.com/articulos/flutter-multiplataforma](https://codigofacilito.com/articulos/flutter-multiplataforma)
- Imlahi Rivas, T. (2017). *Diseño Dirigido por Modelos de Aplicaciones para Dispositivos Móviles*. 108. Recuperado de <http://zaguan.unizar.es/TAZ/EUCS/2014/14180/TAZ-TFG-2014-408.pdf>
- INEC. (2018). *Tecnologías de la Información y Comunicación*.
- INEC. (2019). *Encuesta Nacional sobre las Relaciones Familiares y Violencia de Genero contra las Mujeres (ENVIGMU)*.
- Kniberg, H. (2007). *Scrum and XP from the Trenches (How we do Scrum)*. <https://doi.org/978-1-4303-2264-1>
- Lema Peñafiel, J. C. (2019). *Violencia intrafamiliar en parroquias urbanas del cantón Cuenca*. UNIVERSIDAD DE CUENCA.
- Malpica Velasquez, C. J. (2014). APLICACIÓN DE LA METODOLOGÍA SCRUM PARA INCREMENTAR LA PRODUCTIVIDAD DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA CCJ S.A.C. LIMA PRESENTADA.
- Menzinsky, A., López, G., & Palacio, J. (2019). *Scrum Master* (Iubaris In).
- Muñoz Caiza, Á. E. (2019). *PROYECTO DE REFORMA AL ARTÍCULO 375 DEL CÓDIGO DE LA NIÑEZ Y ADOLESCENCIA PARA GARANTIZAR A LA PERSONA AGRAVIADA EL RESARCIMIENTO DE DAÑOS Y PERJUICIOS*. UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES.
- Murrieta López, R. (2013). *ANÁLISIS AL SISTEMA NORMATIVO SOBRE LA DELICUENCIA ORGANIZADA*.
- Nahuel Delia, L. (2017). Desarrollo de aplicaciones híbridas (Universidad Nacional de la Plata). Recuperado de <https://www.oracle.com/technetwork/es/articles/adf/integracion-continua-adf-parte3-4107412-esa.html>
- Onieva López, J. L. (2018). SCRUM COMO ESTRATEGIA PARA EL APRENDIZAJE COLABORATIVO A TRAVÉS DE PROYECTOS. PROPUESTA DIDÁCTICA PARA SU IMPLEMENTACIÓN EN EL AULA UNIVERSITARIA. *Profesorado, Revista de Currículum y Formación del Profesorado*, 22(2), 509-527. <https://doi.org/10.30827/profesorado.v22i2.7735>
- Palacio, M. (2020). *Scrum Master* (Iubaris Info 4 Media SL, Ed.). Scrum Manager.
- Pascual, I. M. (2013). *Ciberdelincuencia. Desarrollo y persecución tecnológica*. UNIVERSIDAD POLITÉCNICA DE MADRID.
- Passailaigue, D., Galeno, G., González, P., & Lozano, W. (2019). Servicios - Amiga, ya no estas sola. Recuperado 5 de marzo de 2020, de <https://amigayanoestassola.com/servicios/>
- Payano Muñoz, D. J. (2019). *Implementación de la metodología SCRUM para agilizar proceso de atención de requerimientos del departamento de tecnología de la información de la Caja Huancayo*. Huancayo.

- Quintana Zurita, Y., Rosero Moncayo, J., Serrano Salgado, J., & Pimentel Bolaños, J. L. (2014). LA VIOLENCIA DE GÉNEO CONTRA LAS MUJERES EN EL ECUADOR. En *Quito*. Recuperado de [https://www.unicef.org/ecuador/Violencia\\_de\\_Gnero.pdf](https://www.unicef.org/ecuador/Violencia_de_Gnero.pdf)
- Rayón Ballesteros, M. C., & Gómez Hernández, J. A. (2014). Cibercrimen : particularidades en su investigación y enjuiciamiento. *UOC -Universitat Oberta de Catalunya*, 209-234.
- Rodríguez Cerna, R. S. (2018). *Aplicación móvil para la planificación de rutas de transporte público, Chimbote*. Chimbote.
- Silva, M. (2015). *Sistemas Operativos* (Alfaomega, Ed.). Recuperado de [https://books.google.com.ec/books?id=N2MZDQAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q=operativo&f=false](https://books.google.com.ec/books?id=N2MZDQAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q=operativo&f=false)
- Siqueira Simoes, G. (2019). *Fundamentos del modelo SCRUM*. Recuperado de [www.fattocs.com](http://www.fattocs.com)
- Televistazo. (2020, febrero 21). Implementan botón de pánico para mujeres en Guayaquil. Recuperado 5 de marzo de 2020, de Ecuavisa website: <https://www.ecuavisa.com/articulo/noticias/nacional/574821-municipio-guayaquil-implementa-boton-panico-mujeres>
- Torres López, T. M., Munguía Cortés, J. A., & Torres Valdovinos, M. M. (2018). Representaciones sociales de empleo y desempleo en estudiantes universitarios de Quito, Ecuador. *Revista Reflexiones*, 97(2), 20. <https://doi.org/10.15517/rr.v97i2.32704>
- Trigas Gallego, M. (2012). *Metodología Scrum* (Universitat Oberta de Catalunya). Recuperado de <http://hdl.handle.net/10609/17885>
- Tulcán Suquitana, D. G. (2020). *Desarrollo de Aplicaciones Web con el Framework Java Server Faces (JSF)*. UNIVERSIDAD TÉCNICA DEL NORTE, Ibarra.
- Vidal, Vilalta, T. (2019). *Desarrollo de Partyfy, una app híbrida en React Native*. Universitat de Lleida, Lleida.
- Vivar Beltrán, L. A., & Quishpe Sulca, F. G. (2019). *GESTIÓN DEL CONTENIDO MULTIMEDIA DE LA CARTELERA VIRTUAL VÍA STREAMING DE LA ESCUELA DE FORMACIÓN DE TECNÓLOGOS*. ESCUELA POLITÉCNICA NACIONAL.
- Zamora Delgado, R. (2019). EL M-LEARNING, LAS VENTAJAS DE LA UTILIZACIÓN DE DISPOSITIVOS MÓVILES EN EL PROCESO AUTÓNOMO DE APRENDIZAJE. *Revista de Ciencias Humanísticas y Sociales*, 3(9), 29-38. <https://doi.org/10.1017/CBO9781107415324.004>